



aSISt

Moduł mapowania danych / Moduł migawek / Źródła danych

Zespół FINGO

wersja: 5.64.0.0 04.2025

Wszelkie prawa zastrzeżone.

Dokument może być reprodukowany lub przechowywany bez ograniczeń tylko w całości.

Żadna część niniejszego dokumentu, zarówno w całości jak i we fragmentach, nie może być reprodukowana ani przetwarzana w jakikolwiek sposób elektroniczny, fotograficzny, mechaniczny i inny oraz nie może być przechowywana w jakiegokolwiek bazie danych bez pisemnej zgody FINGO sp. z o.o.

Spis treści

1	Wstęp	11
2	Moduł “Mapowanie danych”	13
2.1	Zarządzanie zbiorami.....	14
2.1.1	Tryb dokumenty.....	15
2.1.1.1	Zbiór mapowań.....	15
2.1.1.2	Tabela.....	32
2.1.1.3	Komórka.....	35
2.1.1.4	Widok.....	36
2.1.1.5	Domyślne źródło danych.....	36
2.1.2	Tryb lista.....	37
2.1.2.1	Zbiór mapowań.....	38
2.1.2.2	Widok.....	42
2.2	Zarządzanie źródłami danych.....	42
2.2.1	Źródło danych.....	43
2.2.1.1	Nowe źródło danych.....	43
2.2.1.2	Edytuj źródło danych.....	61
2.2.1.3	Historia źródła danych.....	62
2.2.1.4	Importuj źródła danych.....	62
2.2.1.5	Eksportuj źródła danych.....	63
2.2.1.6	Import zbioru mapowań wraz z używanymi definicjami źródeł.....	64
2.2.1.7	Eksport zbioru mapowań wraz z używanymi definicjami źródeł.....	64
2.2.1.8	Policz wyrażenie.....	65
3	Moduł “Magazyn migawek”	66
3.1	Migawka.....	68
3.1.1	Dodaj na podstawie danych źródłowych.....	68
3.1.2	Dodaj bez danych źródłowych.....	72
3.1.2.1	Dodaj nową tabelę.....	73
3.1.2.2	Usuń tabelę.....	79
3.1.2.3	Edytuj tabelę.....	79
3.1.2.4	Zarządzaj indeksami.....	80
3.1.3	Edytuj definicje migawki.....	83

3.1.3.1	Dodaj nową tabelę.....	84
3.1.3.2	Usuń / przywróć tabelę.....	84
3.1.3.3	Edytuj tabelę	85
3.1.3.4	Zarządzaj indeksami.....	86
3.1.4	Edytuj ETL.....	86
3.1.5	Usuń migawkę.....	87
3.1.6	Importuj migawkę.....	87
3.1.7	Eksportuj migawkę	89
3.1.8	Zarządzaj instancjami migawek.....	89
3.1.8.1	Dodaj nową instancję migawek.....	90
3.1.8.2	Edytuj instancje migawek.....	91
3.2	Widok.....	97
3.2.1	Odśwież.....	97
4	ETL.....	97
4.1	Kroki ETL.....	98
4.1.1	Krok wejściowy	99
4.2	Tworzenie nowego kroku.....	101
4.2.1	Filtrowanie.....	102
4.2.2	Skrypt wiersza.....	104
4.2.2.1	Wykonaj przed.....	106
4.2.2.2	Skrypt.....	106
4.2.2.3	Wykonaj po.....	107
4.2.3	Skrypt tabeli.....	108
4.2.4	Walidacja wiersza.....	111
4.2.4.1	Niepuste kolumny	112
4.2.4.2	Skrypt walidacji wiersza	113
4.2.5	Instrukcja SQL.....	115
4.2.6	Walidacja tabeli.....	117
4.2.6.1	Niepusta tabela	118
4.2.6.2	Zapytanie SQL.....	119
4.2.6.3	Skrypt Groovy.....	120
4.3	Usuwanie kroku.....	123
4.4	Zmiana kolejności prezentowania zdefiniowanych kroków	124
4.5	Walidacja zdefiniowanych kroków.....	124
4.6	Testowe wykonanie zdefiniowanych kroków	125

5	Zasilanie według mapowań.....	125
6	Informacje dodatkowe.....	132
6.1	Źródła danych.....	132
6.1.1	Tabelaryczne źródła danych typu Excel.....	132
6.1.2	Tabelaryczne źródła danych typu CSV.....	132
6.2	Wyrażenia mapujące.....	132
6.2.1	Typy komórek sprawozdawczych.....	133
6.2.2	Rodzaje wyrażeń mapujących.....	135
6.2.2.1	Mapowanie pojedynczych komórek sprawozdawczych.....	136
6.2.2.2	Pobieranie danych ze źródła.....	136
6.2.2.3	Mapowanie komórek listowych.....	138
6.2.2.4	Mapowanie wymiarów użytkownika.....	138
6.2.3	Kontrola poprawności wyrażeń mapujących.....	139
6.2.4	Składnia języka wyrażeń grupujących.....	139
6.2.5	Obliczanie wartości wyrażenia mapującego.....	140
6.2.6	Język tabelarycznych źródeł danych.....	140
6.2.6.1	Gramatyka.....	140
6.2.6.2	Operacje.....	141
6.2.6.3	Operator zbioru.....	143
6.2.6.4	Funkcje manipulujące na wartościach.....	143
6.2.6.5	Funkcje kontekstowe.....	147
6.2.6.6	Funkcje agregujące.....	148
6.2.6.7	Rachunek typów.....	150
6.2.6.8	Funkcje grupujące.....	151
6.2.6.9	Automatyczne formatowanie.....	151
6.2.6.10	Wyrażenia regularne.....	151
6.2.6.11	Obliczanie wyrażeń tabelarycznych.....	152
6.2.6.12	Kolumny wirtualne.....	152
6.2.6.13	Śledzenie przetwarzania tabelarycznego źródła danych.....	153
6.2.6.14	Źródła danych typu Excel komórkowy.....	153
6.2.7	Import i eksport wyrażeń mapujących.....	154
6.2.7.1	Opis elementów zawartych w XML Schema <code></xs:schema></code>	155
6.3	Magazyn migawek.....	157
6.4	ETL.....	158
6.4.1	Tabela wejściowa.....	158
6.4.2	Instrukcja SQL.....	158

6.4.3	Skrypt wiersza.....	158
6.4.4	Skrypt tabeli.....	158
6.4.5	Walidacja wiersza.....	158
6.4.6	Walidacja tabeli.....	159
6.5	Praca w środowisku wielostanowiskowym.....	159

Spis ilustracji

Rysunek 1. Schemat przebiegu procesu sprawozdawczego.....	11
Rysunek 2. Lista zakładek modułowych aSIST – z wyróżnionym modułem „Mapowania danych”	13
Rysunek 3. Moduł mapowania danych – widok ogólny, bez otwartego zbioru mapowań.....	14
Rysunek 4. Menu "Zbiór mapowań" w trybie „Dokumenty”.....	16
Rysunek 5. Tworzenie nowego zbioru wyrażeń mapujących.....	17
Rysunek 6. Komunikat o powtórzeniu nazwy zbioru wyrażeń mapujących, dla wybranego typu taksonomii.....	17
Rysunek 7. Komunikat o pustej nazwie zbioru wyrażeń mapujących.....	17
Rysunek 8. Komunikat o braku zdefiniowanego typu taksonomii dla zbioru wyrażeń mapujących.....	17
Rysunek 9. Zarządzanie zbiorami mapowań – tryb dokumenty – widok z otwartym zbiorem wyrażeń mapujących.....	18
Rysunek 10. Ustawienia zbioru wyrażeń mapujących-dane podstawowe.....	20
Rysunek 11. Ustawienia zbioru wyrażeń mapujących – wymiary użytkownika.....	20
Rysunek 12. Mapowanie wymiarów użytkownika dla sprawozdania PD.....	21
Rysunek 13. Mapowanie wymiarów użytkownika dla sprawozdania LE – wymiar Kod grupy i kod jednostki.....	23
Rysunek 14. Mapowanie wymiarów użytkownika dla sprawozdania LE – wymiar Kod grupy lub kod jednostki.....	24
Rysunek 15. Historia zmian aktywnego zbioru mapowań.....	25
Rysunek 16. Wybór pliku źródłowego i rodzaju importu wyrażeń.....	26
Rysunek 17. Walidacja wyrażeń znajdujących się w pliku do importu	26
Rysunek 18. Zapisywanie poprawnie zaimportowanych wyrażeń.....	27
Rysunek 19. Wybór tabel do eksportu.....	27
Rysunek 20. Wybór pliku docelowego do zapisu wyeksportowanych wyrażeń.....	28
Rysunek 21. Zapisywanie wyeksportowanych niepustych wyrażeń z wybranych tabel.....	28
Rysunek 22. Szczegóły zbioru mapowań.....	29
Rysunek 23. Analiza formularzy przy usuwanie wyrażeń mapujących.....	29
Rysunek 24. Komunikat o braku wyrażeń mapujących	30
Rysunek 25. Prezentacja formularzy z wyrażeniami mapującymi do usunięcia	30
Rysunek 26. Dodatkowe zapytanie czy usunięcie wyrażeń mapujących.....	30
Rysunek 27. Właściwe usuwanie wyrażeń mapujących	31
Rysunek 28. Zamykanie zbioru mapowań	31
Rysunek 29. Menu "Tabela" w trybie „Dokumenty”	32
Rysunek 30. Wybór źródła danych przed generowaniem wyrażeń mapujących.....	33
Rysunek 31. Generacja mapowań dla aktywnej tabeli.....	33

Rysunek 32. Informacja o wygenerowaniu wyrażeń mapujących.....	34
Rysunek 33. Wybór innego niż komórkowe źródło danych, przy wykonywaniu funkcji generowania wyrażeń mapujących.....	34
Rysunek 34. Eksport wyrażeń mapujących do Excela.....	35
Rysunek 35. Informacja o zapisaniu pliku w formacie Excel.....	35
Rysunek 36. Menu "Komórka" w trybie „Dokumenty”.....	35
Rysunek 37. Menu "Widok" w trybie „Dokumenty”.....	36
Rysunek 38. Zarządzanie zbiorami mapowań – lista dostępnych zbiorów wyrażeń mapujących.....	38
Rysunek 39. Funkcje menu „Zbiór mapowań” – tryb „Lista zbiorów mapowań”.....	38
Rysunek 40. Menu "Widok" w trybie „Lista zbiorów mapujących”.....	42
Rysunek 41. Zarządzanie źródłami danych.....	43
Rysunek 42. Menu „Źródło danych" w Zarządzaniu źródłami danych.....	43
Rysunek 43. Definiowanie nowego źródła danych typu CSV.....	46
Rysunek 44. Definiowanie nowego źródła danych typu Excel kolumnowy.....	46
Rysunek 45. Definiowanie nowego źródła danych Excel komórkowy.....	47
Rysunek 46. Definiowanie nowego źródła danych typu Regularne.....	47
Rysunek 47. Definiowanie nowego źródła danych typu Tabela ze słownikami.....	48
Rysunek 48. Definiowanie nowego źródła danych typu JDBC.....	48
Rysunek 49. Definiowanie nowego źródła danych typu „Migawkowe źródło danych”.....	48
Rysunek 50. Definiowanie wewnętrznego źródła danych.....	49
Rysunek 51. Definiowanie nowego źródła danych typu ETL.....	49
Rysunek 52. Parametryzowanie źródła danych typu ETL.....	50
Rysunek 53. Okno prezentujące proces ETL lub migawkę użytą do testowania źródła migawkowego..	50
Rysunek 54. Okno prezentujące proces ETL lub migawkę użytą do testowania źródła ETL.....	51
Rysunek 55. Przetwarzanie procesu ETL lub migawki.....	52
Rysunek 56. Test źródła danych ETL.....	52
Rysunek 57. Typy sprawozdań źródłowych.....	56
Rysunek 58. Wybór okresu względnego.....	57
Rysunek 59. Zarządzanie parametrami przy parametryzowanym źródle danych.....	58
Rysunek 60. Definiowanie nowego parametru dla parametryzowanego źródła danych.....	58
Rysunek 61. Zarządzanie kolumnami wirtualnymi dla źródeł danych.....	59
Rysunek 62. Dodawanie kolumny wirtualnej dla źródeł danych.....	59
Rysunek 63. Formaty dla źródła danych typu Regularne.....	60
Rysunek 64. Historia zmian definicji źródła danych.....	62
Rysunek 65. Import definicji źródeł mapowań.....	63
Rysunek 66. Eksport definicji źródeł mapowań.....	63
Rysunek 67. Eksport zbioru mapowań wraz z używanymi definicjami źródeł.....	64
Rysunek 68. Szybkie sprawdzenie definiowanych wyrażeń do zasilania danymi.....	65

Rysunek 69. Lista zakładek modułowych aSISt – z wyróżnionym modułem „Magazyn migawek”	66
Rysunek 70. Magazyn migawek – prezentacja listy migawek.....	67
Rysunek 71. Menu „Migawka” w module „Magazyn migawek”	68
Rysunek 72. Wybór danych podstawowych przy tworzeniu migawki.....	69
Rysunek 73. Określenie źródeł dla nowotworzonej migawki.....	69
Rysunek 74. Definiowanie źródła plikowego.....	70
Rysunek 75. Dodanie źródła bazodanowego.....	71
Rysunek 76. Komunikat o błędnych danych tworzonego źródła bazodanowego.....	71
Rysunek 77. Wstępna walidacja wybranych danych źródłowych.....	72
Rysunek 78. Prezentacja tabel, z którymi została połączona utworzona migawka.....	72
Rysunek 79. Tworzenie nowej migawki, bez wskazywania danych źródłowych	73
Rysunek 80. Ręczne definiowanie tabel do nowotworzonej migawki	74
Rysunek 81. Konfiguracja źródła danych dla definiowanej tabeli.....	75
Rysunek 82. Informacje o kolumnie definiowanej tabeli.....	76
Rysunek 83. Lista wybranych kolumn dla definiowanej tabeli.....	77
Rysunek 84. Definiowanie migawki – zdefiniowanie i zapisanie wymaganej tabeli.....	77
Rysunek 85. Zarządzanie indeksami.....	80
Rysunek 86. Lista kolumn indeksu	81
Rysunek 87. Zarządzanie indeksami – lista utworzonych indeksów.....	82
Rysunek 88. Edycja definicji migawki.....	84
Rysunek 89. Zapytanie o usunięcie migawki.....	87
Rysunek 90. Wskazanie ścieżki do importowanej migawki	88
Rysunek 91. Prezentacja ustawień importowanej migawki.....	88
Rysunek 92. Informacja o tabelach powiązanych z importowaną migawką.....	88
Rysunek 93. Informacja o zakończeniu eksportu migawki.....	89
Rysunek 94. Lista instancji migawek	90
Rysunek 95. Dane instancji migawki.....	91
Rysunek 96. Informacje szczegółowe prezentowane w instancji migawki.....	91
Rysunek 97. Menu "SQL" w zarządzaniu instancjami migawek.....	92
Rysunek 98. Menu "Lista tabel" w zarządzaniu instancjami migawek.....	93
Rysunek 99. Informacje prezentowane w instancji migawki po wykonaniu zapytania SQL.....	94
Rysunek 100. Historia zapytań SQL.....	95
Rysunek 101. Informacja o wynikach przeprowadzonej walidacji danych w instancji migawek	96
Rysunek 102. Przebieg procesu importu danych do instancji migawek.....	97
Rysunek 103. Menu "Widok" w module „Magazyn Migawek”	97
Rysunek 104. Diagram kroków ETL.....	99
Rysunek 105. ETL – krok wejściowy.....	100
Rysunek 106. ETL – krok wejściowy z widocznym opisem tabeli źródłowej.....	101

Rysunek 107. ETL – funkcje tworzące nowy krok	102
Rysunek 108. ETL – filtrowanie danych w tabelach migawkowych	103
Rysunek 109. Wynik filtrowania.....	104
Rysunek 110. Skrypt wiersza – definicja kroku	105
Rysunek 111. Skrypt wiersza – panel edycji skryptu.....	106
Rysunek 112. Skrypt wiersza – wynik przetwarzania.....	108
Rysunek 113. Skrypt tabeli – definicja kroku.....	108
Rysunek 114. Skrypt tabeli – panel edycji skryptu	109
Rysunek 115. Skrypt tabeli – wynik wykonania.....	111
Rysunek 116. Walidacja wiersza, kolumny – definicja kroku	112
Rysunek 117. Walidacja wiersza, kolumny – zawartość tabeli wejściowej.....	112
Rysunek 118. Walidacja wiersza, kolumny – raport wykonania	113
Rysunek 119. Walidacja wiersza, kolumny – wynik wykonania	113
Rysunek 120. Walidacja wiersza skryptem – definicja kroku.....	114
Rysunek 121. Walidacja wiersza skryptem – panel edycji.....	114
Rysunek 122. Walidacja wiersza skryptem – wynik wykonania.....	115
Rysunek 123. Instrukcja SQL- definicja kroku.....	116
Rysunek 124. Instrukcja SQL – panel edycji.....	116
Rysunek 125. Instrukcja SQL – wynik wykonania.....	117
Rysunek 126. Walidacja, niepusta tabela – definicja kroku.....	117
Rysunek 127. Walidacja, niepusta tabela – definicja kroku.....	118
Rysunek 128. Walidacja, niepusta tabela – raport wykonania	119
Rysunek 129. Walidacja, niepusta tabela – wynik wykonania.....	119
Rysunek 130. Walidacja SQL – definicja kroku.....	119
Rysunek 131. Walidacja SQL – panel edycji.....	120
Rysunek 132. Walidacja SQL – wynik wykonania	120
Rysunek 133. Walidacja Groovy – definicja kroku.....	121
Rysunek 134. Walidacja Groovy – wynik wykonania.....	121
Rysunek 135. Walidacja Groovy – panel edycji.....	121
Rysunek 136. ETL – wyniki walidacji kroków.....	124
Rysunek 137. ETL – lista instancji migawek dostępna przy testowym wykonywaniu zdefiniowanych kroków.....	125
Rysunek 138. Opcje zasilenia danymi po wybraniu funkcji "Zasilaj wg mapowań".....	126
Rysunek 139. Wybór tabel do "Zasilania wg mapowań"	127
Rysunek 140. Prezentacja wykorzystywanych źródeł danych i wybór opcji „śledzenia nieużywanych rekordów”.....	128
Rysunek 141. Prezentacja wykorzystywanych źródeł danych	129
Rysunek 142. Wypełnianie dynamicznych parametrów źródeł danych	129

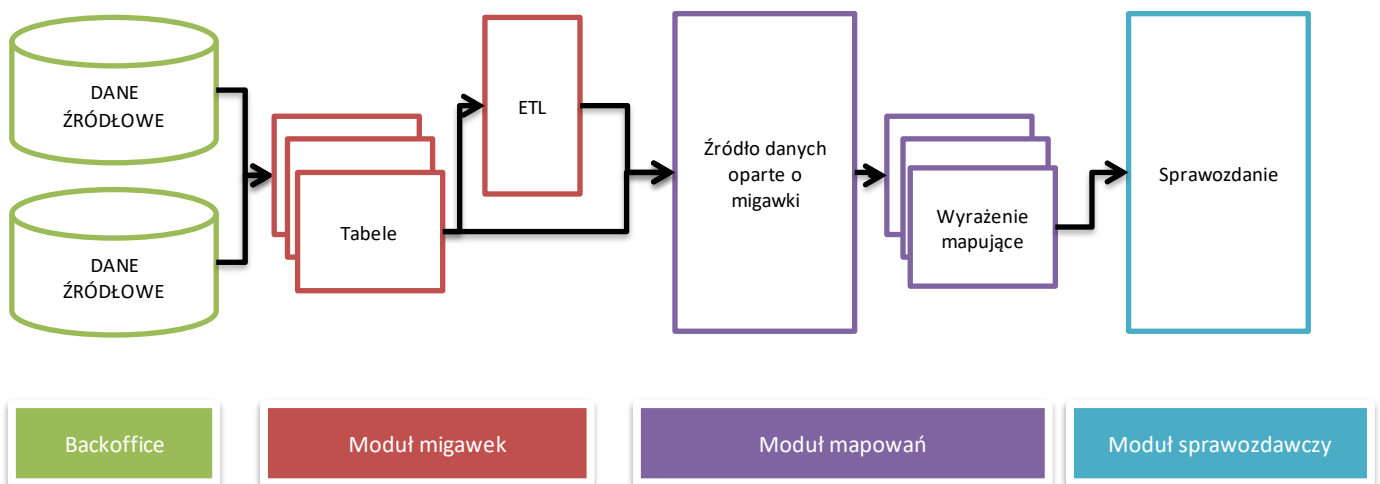
Rysunek 143. Wypełnianie dynamicznych parametrów źródeł danych – brak danych..... 130
 Rysunek 144. Zasilenia wg mapowań – pobieranie danych do zasilenia..... 130
 Rysunek 145. Prezentacja wyników walidacji..... 130
 Rysunek 146. Zapisywanie poprawnych danych do tabel..... 131
 Rysunek 147. Zapisywanie poprawnych danych i nieużywanych rekordów..... 131

1 Wstęp

Aplikacja **aSist** umożliwiając tworzenie różnorodnych sprawozdań finansowych, daje możliwość elastycznego dopasowania się do różnych schematów danych. W tym celu dostarcza zaawansowane mechanizmy pozwalające na pobieranie danych źródłowych z różnych systemów bankowych (tzw. backoffice) i przetwarzanie ich do postaci wymaganej przez nadzorcę.

W zależności od wewnętrznych rozwiązań przyjętych przez bank, dane źródłowe mogą pochodzić z jednego lub z wielu niezależnych systemów (hurtowni danych, systemów transakcyjnych, systemów sprzedaży etc.). W związku z tym, w celu przygotowania końcowego sprawozdania finansowego, konieczne może być np. połączenie danych przed ich dalszym przetworzeniem lub też wykonanie odpowiednich transformacji.

Czynności wykonywane począwszy od pobrania danych z systemu źródłowego do przygotowania raportu końcowego nazywane są procesem sprawozdawczym (zob. Rysunek 1):



Rysunek 1. Schemat przebiegu procesu sprawozdawczego

W pierwszym etapie procesu sprawozdawczego: dane znajdują się w systemach backoffice banku.

Drugi etap, to: przy wykorzystaniu „Modułu migawek”, przeniesienie odpowiednich danych źródłowych do wbudowanego w aplikację aSIS^t: „Repozytorium danych źródłowych”.

W czasie przygotowywania tego etapu może się okazać, że dane będą wymagały transformacji wynikających z konieczności przefiltrowania, agregacji, czy też odpowiedniej selekcji. Aplikacja aSIS^t posiada wbudowany model ETL za pomocą, którego użytkownik może dokonać wszystkich tych operacji.

Przetwarzanie ETL definiowane jest na poziomie „Modułu migawek”.

Trzeci etap, to przy wykorzystaniu „Modułu mapowań”, definiowanie źródeł danych, zbiorów mapowań i wyrażeń mapujących a następnie zasilenie danymi przygotowywanych sprawozdań.

Ostatni etap procesu sprawozdawczego wykonywany jest w „Module sprawozdawczym” i polega na uzgodnieniu raportowanych danych, ich walidacji zgodnie z obowiązującymi regułami kontrolnymi i wygenerowaniu pliku wynikowego.

Moduły: „Mapowania danych” i „Magazyn migawek”, to dodatkowe funkcjonalności aplikacji aSIS^t, które ułatwiają użytkownikom zasilanie przygotowywanych sprawozdań danymi, pochodzącymi w różnorodnych źródłach.

Pozwalają one na:

- tworzenie definicji migawek opartych o tabelaryczne dane oraz z wykorzystaniem wbudowanego w aplikację aSIS^t narzędzie typu ETL (ang. Extract transform and Load) służącego do transformacji danych z systemów źródłowych zasilanie sprawozdań;
- tworzenie wyrażeń mapujących dla dowolnych pól sprawozdawczych, dzięki którym aplikacja aSIS^t przy wykorzystaniu funkcji „Zasilanie wg mapowań”, może zostać zasilona danymi. Źródłem danych podczas wyliczania mapowań mogą być pojedyncze pliki CSV, Excel lub zaawansowane wyniki kroków pochodzących z przekształceń ETL;
- wykonywanie eksportu zdefiniowanych wyrażeń mapujących i ich importu do nowego zbioru;
- tworzenie dowolnej liczby zbiorów do mapowania danych,
- zarządzanie zbiorami do mapowań poprzez ich dezaktywację i usuwanie;
- definiowanie różnych źródeł zasilania i zarządzanie nimi.

Zdefiniowane tu:

- **źródła danych :**
 - są niezależne od obowiązującej taksonomii,
 - mają charakter globalny dla całej aplikacji,
 - mogą być współdzielone przez wykorzystujące je wyrażenia mapujące lub zdefiniowane migawki,

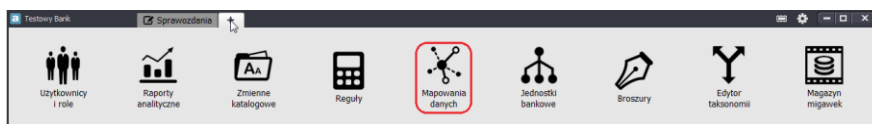
- reprezentują pojęciowo widok na pojedyncze tabelaryczne źródło danych (np. plik CSV, plik MS Excel, wynik zapytania SQL wykonany na danych źródłowych lub kolejnych krokach przetwarzania ETL, etc.);
- **wyrażenia mapujące :**
 - są ściśle powiązane ze „zbiorami wyrażeń mapujących”,
 - dotyczą wszystkich okresów powiązanych z określonym „zbiorem wyrażeń mapujących”, odwołując się do wcześniej zdefiniowanych źródeł danych;
 - wykorzystywane są w ostatniej fazie procesu zasilania,
 - służą do zdefiniowania wartości konkretnej komórki z tworzonego sprawozdania,
- **migawki :**
 - odwołują się do tabelarycznych danych źródłowych, wykorzystywanych w module mapowań jako migawkowe źródła danych;
 - mogą się składać z wielu różnych, niezależnych tabel, których źródłem mogą być np. pliki w formacie csv;
 - nie są powiązane z żadnymi okresami sprawozdawczymi,
 - służą do grupowania do jednego bytu zbioru tabelarycznych danych, które mogą pochodzić z różnych systemów. Zbiór taki może być później wykorzystany przy definiowaniu źródła danych.

2 Moduł „Mapowanie danych”



Aby uruchomić moduł „Mapowania danych”, należy wybrać:



lub



Rysunek 2. Lista zakładek modułowych aSIST – z wyróżnionym modułem „Mapowania danych”

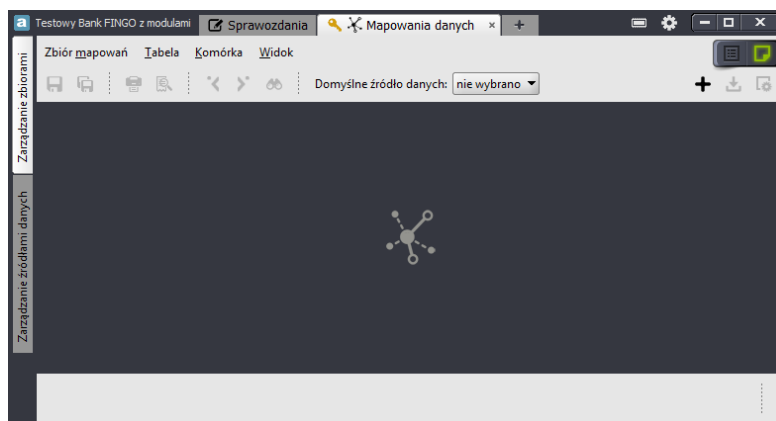
Na pasku zakładek modułowych utworzy się wówczas nowa zakładka  **Mapowania danych**  (zob. Rysunek 3), grupująca funkcje pozwalające na:

- tworzenie dowolnej ilości zbiorów wyrażeń mapujących i zarządzanie nimi;
- definiowanie wyrażeń mapujących i źródeł danych;
- zarządzanie źródłami danych do mapowań,
- wykonanie eksportu zdefiniowanych wyrażeń mapujących i ich importu do innego zbioru wyrażeń mapujących.

Moduł „Mapowania danych”:

- dostępny jest wyłącznie dla banków, posiadających licencję rozszerzoną o ten moduł;
- przy wersji wielostanowiskowej może być obsługiwany jednocześnie wyłącznie na jednym stanowisku.

Po wybraniu funkcji „Moduł mapowania danych”, na ekranie monitora widoczny jest obraz (zob. Rysunek 3):



Rysunek 3. Moduł mapowania danych - widok ogólny, bez otwartego zbioru mapowań

Wszystkie możliwe do wykonania funkcje zostały tu pogrupowane w dwóch zakładkach bocznych:

- zarządzanie zbiorami (zob. rozdział [Zarządzanie zbiorami](#)),
- zarządzanie źródłami danych (zob. rozdział [Zarządzanie źródłami danych](#)).

2.1 Zarządzanie zbiorami

Uaktywnienie modułu „Mapowanie danych” i wybór bocznej zakładki „Zarządzanie zbiorami” (zob. Rysunek 3), pozwala na:

- **tworzenie nowych i zarządzanie istniejącymi zbiorami wyrażeń mapujących**, poprzez ich:
 - otwieranie,
 - deaktywowanie,
 - aktywowanie wcześniej deaktywowanych,
 - sprawdzanie ustawień i szczegółów;
 - przeglądanie historii zmian,
- **definiowanie wymaganych wyrażeń mapujących**, dla wcześniej utworzonych zbiorów mapowań, poprzez:
 - wprowadzanie nowych i usuwanie istniejących wyrażeń;
 - eksport i import wyrażeń mapujących.

Funkcje te, odpowiednio pogrupowane, dostępne są w dwóch trybach:

- w trybie „Dokumenty” – odnoszącym się do otwartych zbiorów mapowań,
 - w trybie „Listy” – dotyczących wcześniej utworzonych zbiorów mapowań.
- Niektóre z funkcji dostępne są w obu trybach a inne tylko w jednym z nich.

W zależności od wybranego trybu pracy, „Przełącznik widoku dokumentów”, znajdujący się w prawym górnym rogu ekranu, może mieć postać:



– gdy włączony jest tryb „Dokumenty”




– gdy włączony jest tryb „Lista”

Przełączanie między trybami odbywa się automatycznie, po:

- kliknięciu na wyciemnioną część klawisza funkcyjnego lub
- wykonaniu niektórych funkcji.

Szczegółowe omówienie przełączników widoku, które dla większości zakładki modułowych zostały włączone do aplikacji aSIS od wersji 5.0 i ułatwiają szybkie przełączanie między dwoma trybami pracy, omówione zostało w instrukcji [aSIS-Podstawy obsługi aplikacji.pdf](#) w rozdziale „Przełączniki widoku dokumentów”.

2.1.1 Tryb dokumenty

Tryb „Dokumenty”, dla którego przełącznik widoku dokumentów ma postać:  to tryb, w którym wszystkie możliwe do wykonania funkcje (zob. Rysunek 3) pogrupowane zostały w menu:

- Zbiór mapowań,
- Tabela,
- Komórka,
- Widok

oraz dodatkowo, udostępniona została opcja:

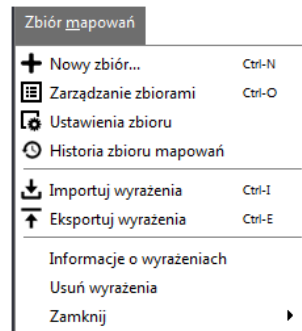
- Domyślny wybór źródła danych.

2.1.1.1 Zbiór mapowań

Menu „Zbiór mapowań” (zob. Rysunek 4), grupuje podstawowe funkcje związane ze zbiorami mapującymi, pozwalające na:

- tworzenie i zarządzanie zbiorami wyrażeń mapujących;
- wprowadzanie wyrażeń mapujących dla dowolnych komórek (w tym tworzenie wyrażeń mapujących dla wymiarów użytkownika),

- import i eksport wyrażeń mapujących;
- usuwanie nieaktualnych wyrażeń mapujących,
- przeglądanie historii czynności wykonywanych w zbiorach mapujących.



Rysunek 4. Menu "Zbiór mapowań" w trybie „Dokumenty”

2.1.1.1 Nowy zbiór wyrażeń mapujących

Aby utworzyć „Nowy zbiór wyrażeń mapujących”, należy wybrać:

- Menu „Zbiór mapowań” →  Nowy zbiór...



lub

- Ikonę: 

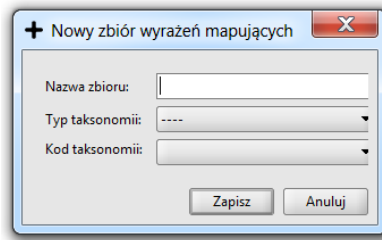
lub

- skrót klawiszowy: Ctrl-N

Funkcja ta:

- dostępna jest w obu trybach pracy, czyli:
 - w trybie „Dokumenty” -  ,
 - w trybie „Lista” -  ,
- pozwala tworzyć nowe zbiory mapowania danych, w których definiowane będą wyrażenia mapujące;
- wymaga tworzenia odrębnych zbiorów mapowania dla różnych typów taksonomii,
- traktuje tworzone zbiory wyrażeń mapujących jako zbiory wzajemnie niezależne, wykorzystywane następnie przez funkcję „Zasilanie według mapowań” do zasilania sprawozdań danymi zewnętrznymi, zgodnie ze zdefiniowanym źródłem danych.

Po uruchomieniu tej funkcji, wyświetla się ekran (zob. Rysunek 5):



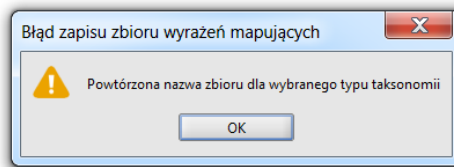
Rysunek 5. Tworzenie nowego zbioru wyrażeń mapujących

Aby utworzyć nowy zbiór wyrażeń mapujących, należy wprowadzić:

- **nazwę zbioru** – unikatowa nazwa, zawierająca maksymalnie 20 dowolnych znaków;
- **typ taksonomii** – wybrać właściwy rodzaj sprawozdania dostępny z wyświetlonej tu listy, dla którego definiowane będą wyrażenia mapujące;
- **kod taksonomii** – kod, istniejący w bazie aSIS^t, wybrany spośród wyświetlonej listy.

Przy wprowadzaniu nazwy zbioru należy pamiętać, że:

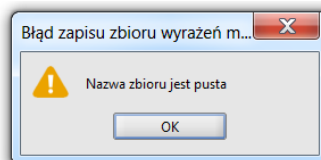
- nie można wprowadzić takiej samej nazwy dla zbiorów o tym samym typie taksonomii, gdyż zbiór taki nie utworzy się (zob. Rysunek 6);



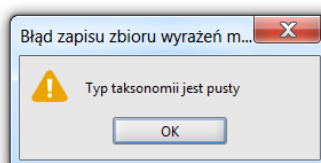
Rysunek 6. Komunikat o powtórzeniu nazwy zbioru wyrażeń mapujących, dla wybranego typu taksonomii

- taka sama nazwa może zostać wprowadzona tylko dla różnych typów taksonomii.


Wymagane jest wypełnienie wszystkich pól, gdyż inaczej nowy zbiór nie zostanie utworzony, o czym zostaniemy poinformowani stosownym komunikatem (zob. Rysunek 7 i Rysunek 8):

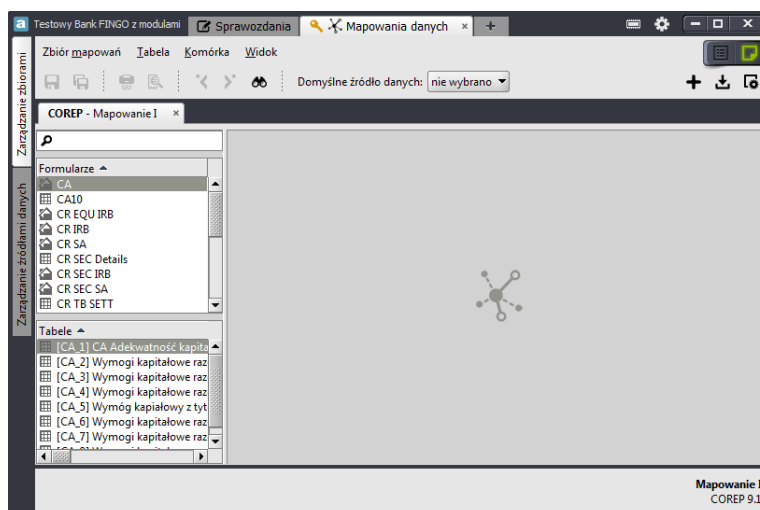


Rysunek 7. Komunikat o pustej nazwie zbioru wyrażeń mapujących



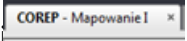

Rysunek 8. Komunikat o braku zdefiniowanego typu taksonomii dla zbioru wyrażeń mapujących

Po poprawnym wprowadzeniu wszystkich wymaganych danych i zatwierdzeniu klawiszem , utworzy się i uaktywni nowy wymagany zbiór do mapowań, a ekran modułu „Mapowania danych” (zob. Rysunek 9 i por. Rysunek 3).



Rysunek 9. Zarządzanie zbiorami mapowań – tryb dokumenty – widok z otwartym zbiorem wyrażeń mapujących

UTWORZONY ZBIÓR WYRAŹEŃ MAPUJĄCYCH:

- prezentuje tabele zgodne z wybranym typem i kodem taksonomii, określonym podczas definiowania zbioru;
- posiada aktywne wszystkie tabele dla danego typu taksonomii, niezależnie od tego, które z nich są przez bank sporządzane;
- obok nadanej nazwy, posiada informację o tym z jakim typem taksonomii został powiązany utworzony zbiór (zob. Rysunek 9):
 - w formie zakładki ,
 - prawym rogu ekranu. ,
- pozwala na definiowanie i zapisywanie nowych wyrażeń mapujących, a później ich przeglądanie i edytowanie;
- prezentuje wszystkie tabele w analogiczny sposób jak po utworzeniu sprawozdania w module „Sprawozdania”, jednak tutaj do poszczególnych pól tabel wprowadzane są nie dane, lecz wyrażenia mapujące, które przy wykorzystaniu funkcji „Zasilanie wg mapowań” pozwolą na zasilenie sprawozdań danymi zewnętrznymi, zgodnie ze zdefiniowanym źródłem danych.

W module „Mapowania danych” może być otwartych jednocześnie wiele zbiorów mapujących i wówczas przełączanie się między nimi odbywa się tak jak przy normalnych sprawozdaniach poprzez kliknięcie na wybrany zbiór.

Aktywny jest zawsze ten zbiór, którego nazwa prezentowana jest w zakładce o jasnym tle, czyli:

 - aktywny zbiór mapowań,

 - nieaktywny zbiór mapowań,

Aby prawidłowo zdefiniować wyrażenie mapujące do zasilenia danymi, należy:

- otworzyć właściwy ZBIÓR WYRAŻEŃ MAPUJĄCYCH,
- otworzyć w nim tabelę, dla której mają zostać wprowadzone wyrażenia mapujące;
- ustawić kursor w komórce, do której mają być zapisane wyrażenie mapujące;
- w zależności od pierwotnego typu komórki, dla której tworzone jest wyrażenie mapujące, wprowadzić odpowiedni typ formułowanego wyrażenia (zob. rozdział [Wyrażenia mapujące](#)).

Powyżej opisany sposób postępowania przy tworzeniu wyrażeń mapujących jest właściwy dla wszystkich pól tabel sprawozdawczych z wyjątkiem „Wymiarów użytkownika”.

Tworzenie wyrażeń mapujących dla „Wymiarów użytkownika” wykonywane jest przy wykorzystaniu funkcji „Ustawienia” (zob. rozdział [Ustawienia zbioru wyrażeń mapujących](#)).

2.1.1.1.2 Zarządzanie zbiorami

Aby przejść do „Zarządzania zbiorami mapowań”, należy wybrać:

- **Menu „Zbiór mapowań”** → [Zarządzanie zbiorami](#)

lub

- jeden ze skrótów klawiszowych: Ctrl-O lub Ctrl-I
bądź

- przełączyć tryb pracy na „Listę”  .

Funkcje pogrupowane w „Zarządzaniu zbiorami mapowań” (zob. rozdział [Tryb lista](#)).

2.1.1.1.3 Ustawienia zbioru wyrażeń mapujących

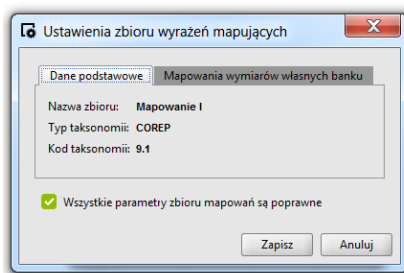
Aby uruchomić funkcję „Ustawienia zbioru mapowań”, należy wybrać:

- **Menu „Zbiór mapowań”** →  [Ustawienia zbioru](#)

Funkcja ta:

- przedstawia podstawowe informacje o otwartym – aktywnym zbiorze wyrażeń mapujących,
- dla zbiorów, w których użytkownik definiuje wymiary, pozwala tworzyć wyrażenia mapujące dla tych wymiarów;

Po uruchomieniu tej funkcji, wyświetla się ekran (zob. Rysunek 10):



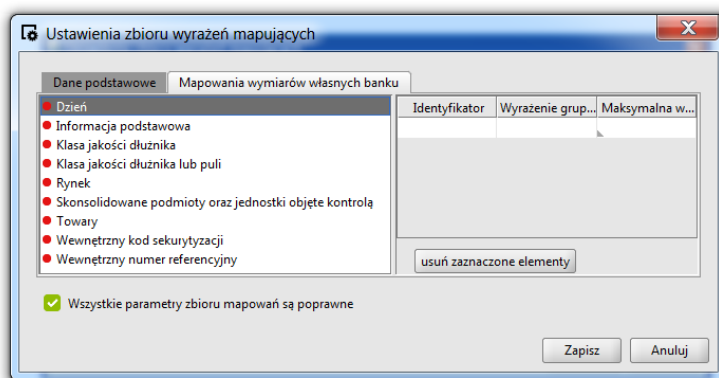
Rysunek 10. Ustawienia zbioru wyrażeń mapujących-dane podstawowe

W zakładce **Dane podstawowe** (zob. Rysunek 10):

- prezentowane są informacje o aktywnym zbiorze wyrażeń mapujących, które zostały wprowadzone przy jego tworzeniu, czyli:
 - nazwa zbioru,
 - typ taksonomii
 - kod taksonomii

Zakładka **Mapowania wymiarów własnych banku** (zob. Rysunek 11):

- w lewym panelu okna, przedstawia wszystkie możliwe do definiowania przez bank wymiary użytkownika
- w prawym panelu okna, pozwala na wprowadzanie wyrażeń mapujących dla wybranego wymiaru użytkownika



Rysunek 11. Ustawienia zbioru wyrażeń mapujących - wymiary użytkownika

Dla różnych wymiarów użytkownika, prawy panel okna może prezentować różną ilość kolumn, do których należy wprowadzić odpowiednie wyrażenia mapujące.

W prawym panelu okna prezentowane są:

- **Identyfikator**, dla którego wartość nadawana jest automatycznie przez aplikację,
- **Wyrażenie grupujące**, czyli pole, do którego należy wprowadzić wyrażenie, odnoszące się do posiadanego źródła danych i służące do wyodrębnienia grup, w obrębie których powstają osobne

wartości. Dla każdej kombinacji wartości powstaną nowe grupy będące podstawą dla obliczenia wartości wymiaru i należących do niego komórek sprawozdania.

Uproszczone składnia takiego wyrażenia, to:

<symbol_źródła>: { <funkcja_grupująca> (<wyrażenie_grupujące >) }

- **Pozostałe kolumny**, do których wprowadza się wyrażenia obliczające wartości elementów poszczególnych wymiarów użytkownika podczas zasilania danymi. Jest ich tyle, ile jest elementów dla danego wymiaru użytkownika.

O tym, czy do wymiaru prezentowanego w lewym panelu okna zostało wprowadzone wyrażenie mapujące czy też nie, informuje kolor kropki znajdujący się obok każdego wymiaru i tak:

- - oznacza, że do danego wymiaru zostało wprowadzone wyrażenie mapujące;
- - oznacza, że wymiar nie posiada w pełni zdefiniowanych wyrażen mapujących.

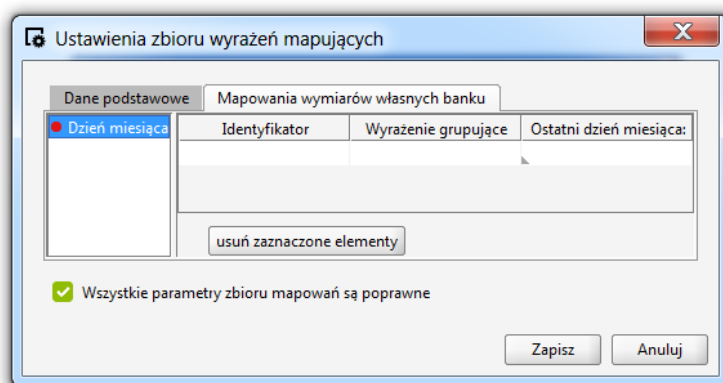
Mapowanie wymiarów użytkownika dla PD

Aby w sprawozdaniu PD prawidłowo zmapować wymiary użytkownika, którymi są dni miesiąca, należy wybrać:

- **Menu „Zbiór mapowań”** →  Ustawienia zbioru ,
- zakładka **Mapowania wymiarów własnych banku** .

I po wybraniu:

- **w lewym panelu** okna wymiaru „Dzień”,
- **w prawym panelu** okna (zob. Rysunek 12) wprowadzić odpowiednie wyrażenia mapujące.



Rysunek 12. Mapowanie wymiarów użytkownika dla sprawozdania PD

- **Wyrażenie grupujące** – wyrażenie, które dla każdej daty (każdego dnia) wyodrębni osobną grupę.

Wyrażenie to może mieć postać:

pd:{ group(\$DATA_DANYCH) }

gdzie:

pd – identyfikator źródła danych,

DATA_DANYCH – kolumna, w której znajdują się daty

Jeżeli z naszego źródła danych, w którym znajdują się dane z różnych miesięcy, chcemy wyodrębnić :

- tylko te dotyczące np. września, to utworzone tu wyrażenie może mieć postać:

pd:{ conditionGroup(getMonth(\$DATA_DANYCH) = 9, toDate(\$DATA_DANYCH)) }

- dane dotyczące przygotowywanego okresu sprawozdawczego, to utworzone tu wyrażenie może mieć postać:

pd:{ conditionGroup(getMonth(\$DATA_DANYCH) = getMonth(getReportDate()), \$DATA_DANYCH) }

warunek:

getMonth(\$DATA_DANYCH) = getMonth(getReportDate())

sprawdza czy miesiąc daty jest miesiącem sprawozdania.

- **Wyrażenie obliczające dzień miesiąca**

Grupy można wyobrazić sobie jako osobne źródła danych dla każdej wartości grupy.

W związku z tym, że określona grupa zawiera jedynie dane dla jednego dnia – oczywiście takich wierszy może być więcej niż jeden. Wartość wymiaru jest dniem z takiej daty, czyli:

getDay(\$DATA_DANYCH)

funkcja: getDay zwraca dzień z daty

Niestety wartości wymiaru dzień posiadają zero na początku dla dni 1-9, czyli 01,02... co oznacza to, że trzeba dodatkowo skomplikować wyrażenie, aby ująć takie przypadki.

Właściwe wyrażenie w 3 kolumnie powinno mieć postać:

pd:{ concat(case(getDay(\$DATA_DANYCH) <= 9, '0', ''), toString(getDay(\$DATA_DANYCH))) }

concat to funkcja sklejjąca dwa napisy – concat('0', '1') to 01 a concat("", '12') to 12

case to funkcja decydująca czy dokleić pusty napis czy '0' w zależności od wartości dnia.

Tak przygotowane mapowanie dla wymiaru pozwoli zmapować poszczególne komórki w tabeli. Przykładowe mapowanie będzie zatem wyliczane niezależnie dla każdego dnia.


Na przykład dla zobowiązań bieżących w PLN może wyglądać tak:

```
pd:{ sum( $ZOB_BIEZ, $SWALUTA = PLN ) }
```

W zasilonym sprawozdaniu dla każdego dnia suma kolumny ZOB_BIEZ będzie policzona osobno, w przypadku, kiedy określony dzień nie będzie miał żadnych rekordów w źródle pd to wyliczona wartość będzie równa 0.

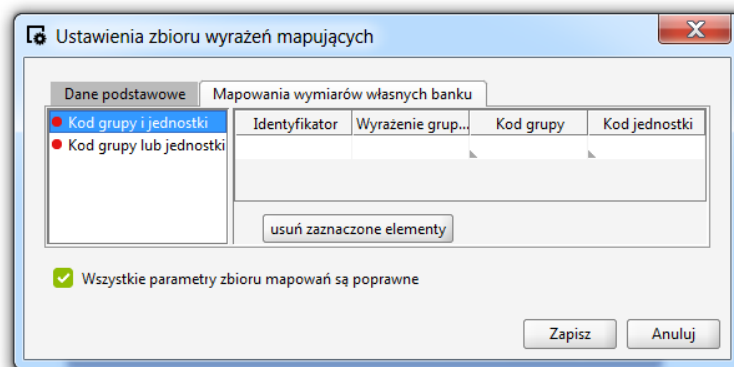
Mapowanie wymiarów użytkownika dla LE

Aby w sprawozdaniu LE prawidłowo zmapować wymiary użytkownika, którymi są „Kod grupy i kod jednostki” oraz „Kod grupy lub Kod jednostki”, należy wybrać:

- Menu „Zbiór mapowań” →  Ustawienia zbioru ,
- a następnie
- zakładka  Mapowania wymiarów własnych banku

i dla odpowiednich wymiarów wybranych z lewego panelu okna:

- w prawym panelu okna (zob. Rysunek 13) wprowadzić odpowiednie wyrażenia mapujące



Rysunek 13. Mapowanie wymiarów użytkownika dla sprawozdania LE - wymiar Kod grupy i kod jednostki

Dla wymiaru „Kod grupy i kod jednostki” (zob. Rysunek 12):

- **Wyrażenie grupujące** - to wyrażenie, które pozwoli ze źródła danych wyodrębnić wszystkie grupy złożone z „Kodu grupy” i „Kodu jednostki”.

Zdefiniowane tu wyrażenie może mieć postać:

```
LE:{ conditionGroup( $KOD_GRUPY != "", $KOD_GRUPY, $KOD_JEDN ) }
```

gdzie:

LE – źródło danych

\$KOD_GRUPY – kolumna, w której znajdują się kody grup

\$KOD_JEDN – kolumna, w której znajdują się kody jednostek

Tak zdefiniowane grupowanie oznacza, że dla każdej kombinacji wartości KOD_GRUPY i KOD_JEDNOSTKI zostanie utworzona osobna grupa, a dla niej określona wartość wymiaru.

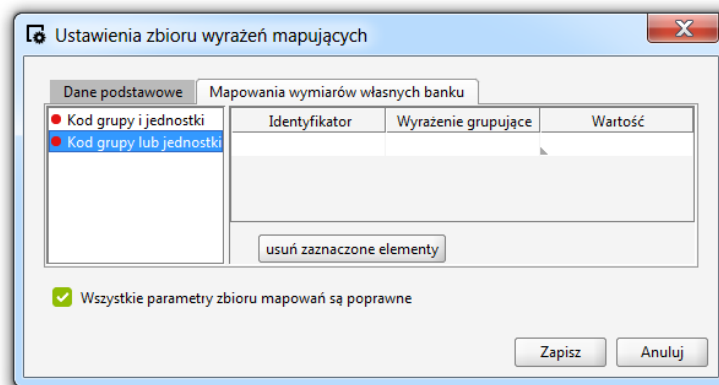
- **Kod grupy** – to wyrażenie obliczające wartość wymiaru „Kod grupy”, które może mieć postać:

LE: { \$KOD_GRUPY }

- **Kod jednostki** – to wyrażenie obliczające wartość wymiaru „Kod jednostki”, które może mieć postać:

LE: { \$KOD_JEDN }

Dla wymiaru „Kod grupy lub kod jednostki” (zob. Rysunek 14):



Rysunek 14. Mapowanie wymiarów użytkownika dla sprawozdania LE - wymiar Kod grupy lub kod jednostki

- **Wyrażenie grupujące** – to wyrażenie, które ze źródła danych wyodrębni wszystkie grupy złożone z „Kodu grupy” lub „Kodu jednostki”.

Wyrażenie tu zdefiniowane może mieć postać:

LE: { group(case(\$KOD_GRUPY != "", \$KOD_GRUPY, \$KOD_JEDN)) }

W tym przypadku dla rekordów w źródle posiadających niepusty KOD_GRUPY grupa jest wyznaczana na podstawie kolumny KOD_GRUPY, lub na podstawie kolumny KOD_JEDNOSTKI dla pozostałych rekordów.

- **Wartość** – to wyrażenie, obliczające wartość „Kodu grupy” lub „Kodu jednostki”, które może mieć postać:

LE: { case(\$KOD_GRUPY != "", \$KOD_GRUPY, \$KOD_JEDN) }

Tak zdefiniowane mapowania wymiarów użytkownika pozwalają wyliczać wartości na poszczególnych tabelach wykonując wyrażenia mapujące wielokrotnie dla każdej wartości wymiaru, która wynika z danych w źródle.

2.1.1.1.4 Historia zbioru mapowań

Aby uruchomić funkcję „Historia zbioru mapowań”, należy wybrać:

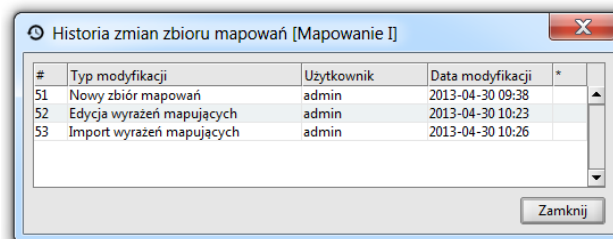
- **Menu „Zbiór mapowań”** →  Historia zbioru mapowań

lub

- Ikonę: .

Funkcja ta (zob. Rysunek 15), przedstawia:

- historię zmian wprowadzonych do aktywnego zbioru mapowań.



Rysunek 15. Historia zmian aktywnego zbioru mapowań

2.1.1.1.5 Importuj wyrażenia mapujące

Aby uruchomić funkcję „Import wyrażeń mapujących”, należy wybrać:

- **Menu „Zbiór mapowań”** →  Importuj wyrażenia

lub

- Ikonę: 

lub

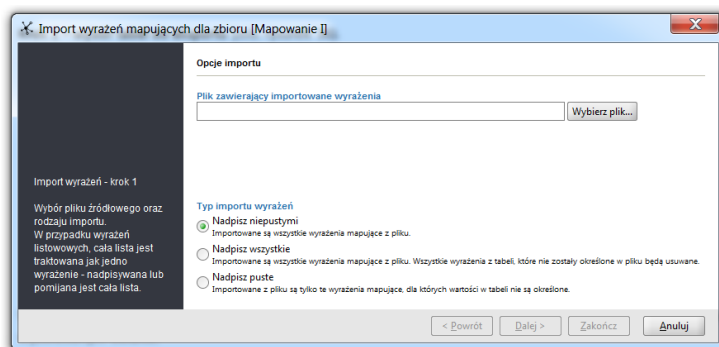
- skrót klawiszowy: Ctrl-I.

Funkcja ta:

- pozwala na zaimportowanie wcześniej zdefiniowanych wyrażeń mapujących.

Pełny import wyrażeń mapujących, każdorazowo przebiega w trzech krokach.

Krok 1 – wybór pliku źródłowego i rodzaju importu (zob. Rysunek 16).



Rysunek 16. Wybór pliku źródłowego i rodzaju importu wyrażen

Typy importu wyrażen:

Nadpisz niepuszymi – spowoduje:

- zaimportowanie wszystkich wyrażen mapujacych z pliku zewnetrznego,
- zachowanie tylko tych wzczesniej wprowadzonych wyrazen mapujacych, ktore nie zostaly ujęte w importowanym pliku.

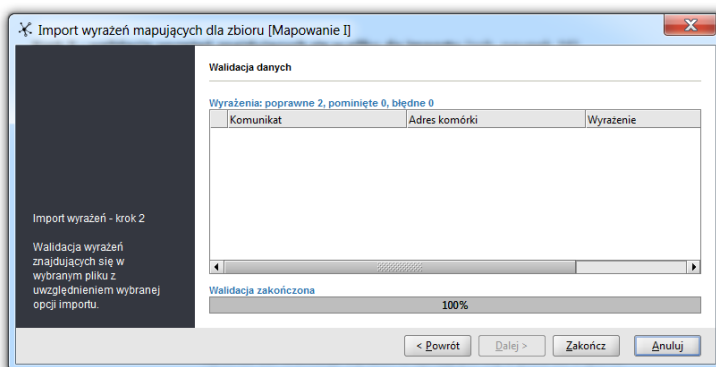
Nadpisz wszystkie – spowoduje:

- zaimportowanie wszystkich wyrażen mapujacych z pliku zewnetrznego,
- usuniecie wszystkich wyrazen mapujacych, ktore zostaly wprowadzone przed importem (dot. tych sprawozdan, do ktorych importowane beda wyrazenia z pliku zewnetrznego).

Nadpisz puste – spowoduje:

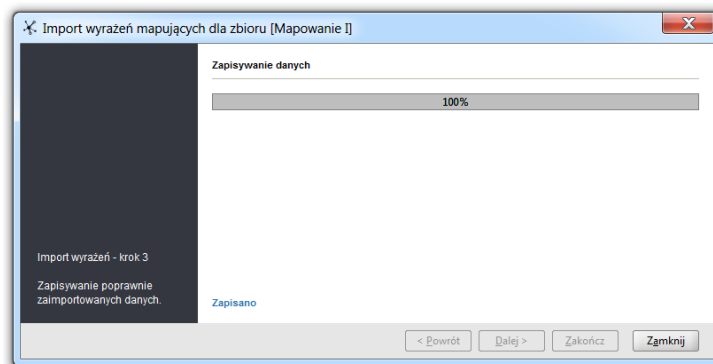
- zaimportowanie tylko tych wyrazen mapujacych z pliku zewnetrznego, ktore w tabelach sprawozdawczych nie mialy wprowadzonych zadnych wartosci;
- zachowanie wszystkich wzczesniej zapisanych wyrazen mapujacych.

Krok 2 – walidacja wyrazen znajdujacych sie w pliku do importu (zob. Rysunek 17).



Rysunek 17. Walidacja wyrażen znajdujacych sie w pliku do importu



Krok 3 – zapisywanie poprawnie zaimportowanych wyrazen (zob. Rysunek 18).



Rysunek 18. Zapisywanie poprawnie zaimportowanych wyrażeń

2.1.1.1.6 Eksportuj wyrażenia mapujące

Aby uruchomić funkcję „Eksportuj wyrażenia mapujące” należy wybrać:

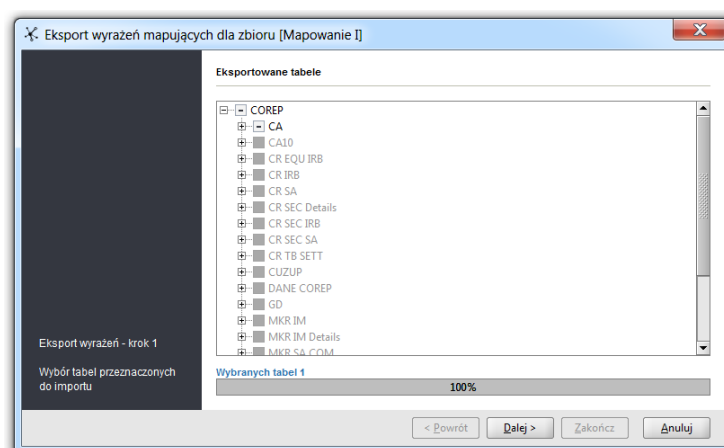
- Menu „Zbiór mapowań” →  Eksportuj wyrażenia ,
 - Ikonę: 
- bądź
- skrót klawiszowy: Ctrl-E.

Funkcja ta:

- pozwala z aktywnego zbioru mapowań, na wyeksportowanie zdefiniowanych tam wyrażeń mapujących,
- umożliwia wykonanie eksportu mapowań z wszystkich lub tylko niektórych tabel,
- zapisuje wyeksportowane dane w formacie XML lub XLSX.

Pełny proces eksportu wyrażeń mapujących wykonywany jest w trzech krokach.

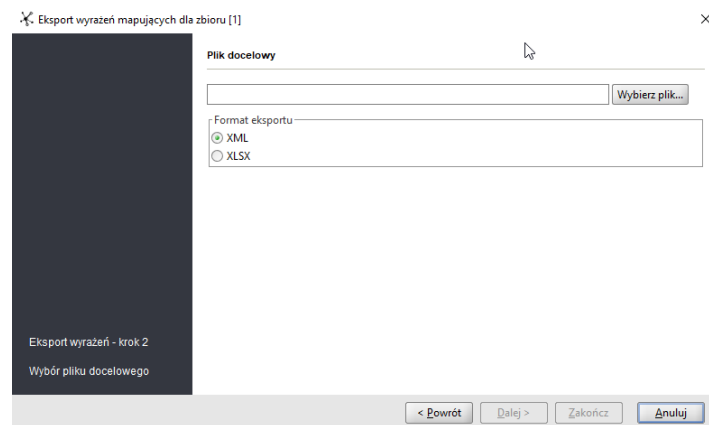
Krok 1 – wybór tabel do eksportu (zob. Rysunek 19)



Rysunek 19. Wybór tabel do eksportu

Wyeksportowane będą tylko te wcześniej zdefiniowane wyrażenia, których tabele zaznaczone są na wyświetlonym ekranie.

Krok 2 – wybór pliku docelowego z wyeksportowanymi wyrażeniami (zob. Rysunek 20).

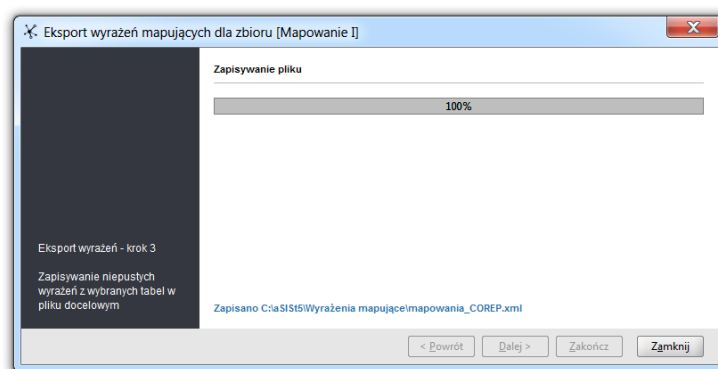


Rysunek 20. Wybór pliku docelowego do zapisu wyeksportowanych wyrażeń


Wymagane jest tutaj:

- wybranie ścieżki do zapisu wyeksportowanych wyrażeń mapujących,
- podanie nazwy tworzonego pliku.

Krok 3 – zapisywanie niepustych wyrażeń z wybranych tabel (zob. Rysunek 21).



Rysunek 21. Zapisywanie wyeksportowanych niepustych wyrażeń z wybranych tabel

Po zapisaniu wyeksportowanych wyrażeń, wyjście z tej funkcji następuje po wybraniu przycisku .

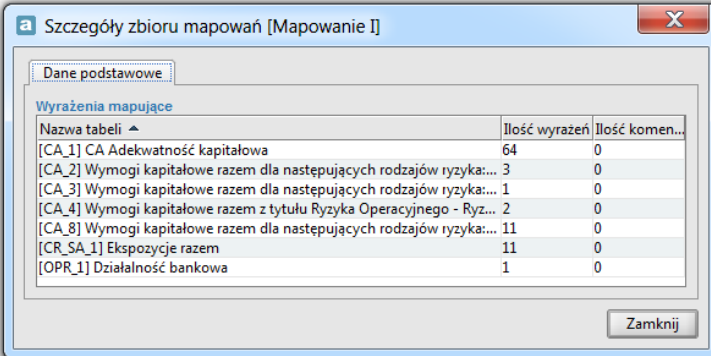
2.1.1.1.7 Informacje o wyrażeniach

Aby uruchomić funkcję „Informacja o wyrażeniach zbioru mapowań”, należy wybrać:

- **Menu „Zbiór mapowań”** → *Szczegóły zbioru*

Funkcja ta (zob. Rysunek 22), prezentuje:

- listę tabel, w których zostały zdefiniowane wyrażenia mapujące;
- liczbę zawartych wyrażeń i komentarzy dla tych tabel.



Nazwa tabeli	Ilość wyrażeń	Ilość komen...
[CA_1] CA Adekwatność kapitałowa	64	0
[CA_2] Wymogi kapitałowe razem dla następujących rodzajów ryzyka:...	3	0
[CA_3] Wymogi kapitałowe razem dla następujących rodzajów ryzyka:...	1	0
[CA_4] Wymogi kapitałowe razem z tytułu Ryzyka Operacyjnego - Ryz...	2	0
[CA_8] Wymogi kapitałowe razem dla następujących rodzajów ryzyka:...	11	0
[CR_SA_1] Ekspozycje razem	11	0
[OPR_1] Działalność bankowa	1	0

Rysunek 22. Szczegóły zbioru mapowań

2.1.1.1.8 Usuń wyrażenia mapujące

Aby uruchomić funkcję „Usuń wyrażenia mapujące”, należy wybrać:

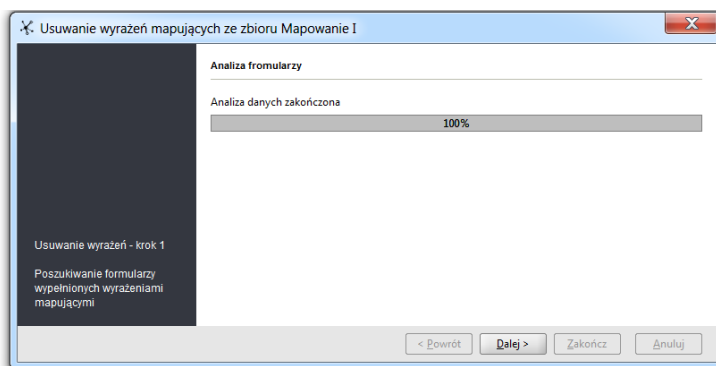
- **Menu „Zbiór mapowań”** → **Usuń wyrażenia**

Funkcja ta:

- pozwala na usunięcie wszystkich wyrażeń z aktywnego zbioru mapującego.

Pełny proces usuwania wyrażeń mapujących wykonywany jest w trzech krokach.

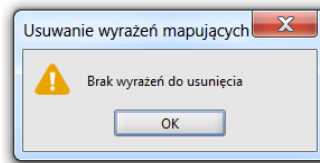
Krok 1 – poszukiwanie formularzy wypełnionych wyrażeniami mapującymi (zob. Rysunek 23).



Rysunek 23. Analiza formularzy przy usuwanie wyrażeń mapujących

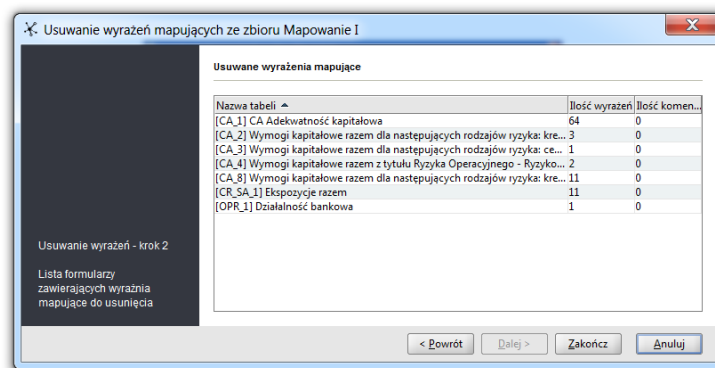
Następuje tu przeszukiwanie wszystkich formularzy aktywnego zbioru mapującego celem sprawdzenia, czy znajdują się tam formularze wypełnione wyrażeniami mapującymi.

W przypadku braku takich formularzy, jesteśmy o tym informowani komunikatem (zob. Rysunek 24).



Rysunek 24. Komunikat o braku wyrażeń mapujących

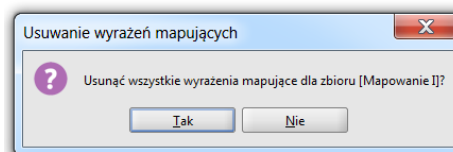
Krok 2 – prezentacja formularzy zawierających wyrażenia mapujące do usunięcia (zob. Rysunek 25).



Rysunek 25. Prezentacja formularzy z wyrażeniami mapującymi do usunięcia

Na ekranie zostaje przedstawiona lista wszystkich formularzy zawierająca wyrażenia mapujące, które zostaną usunięte, jeśli proces usuwania będzie kontynuowany.

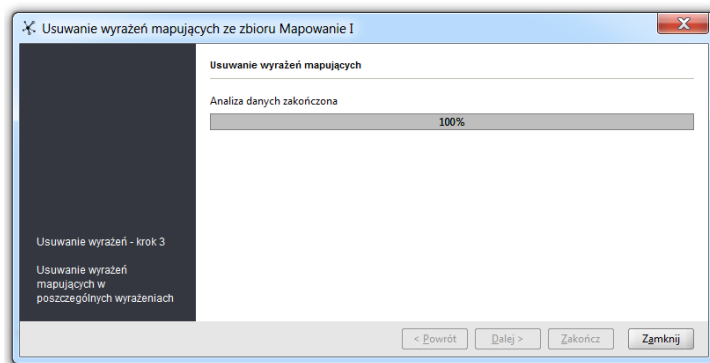
Wybranie przycisku , spowoduje wyświetlenie dodatkowego zapytania (zob. Rysunek 26)




Rysunek 26. Dodatkowe zapytanie czy usunięcie wyrażeń mapujących

Po pozytywnym zatwierdzeniu tego komunikatu, nastąpi usunięcie wszystkich wyrażeń mapujących z wcześniej przedstawionych formularzy.

Krok 3 – właściwe usuwanie wyrażeń mapujących (zob. Rysunek 27).



Rysunek 27. Właściwe usuwanie wyrażeń mapujących

Wybranie przycisku , zakończy proces usuwania wyrażeń mapujących i aktywny zbiór mapowań nie będzie zawierał już żadnych, wcześniej zdefiniowanych mapowań.

2.1.1.1.9 Zamknij zbiór mapowań



Aby uruchomić funkcję „Zamknij zbiór mapowań”, należy wybrać:

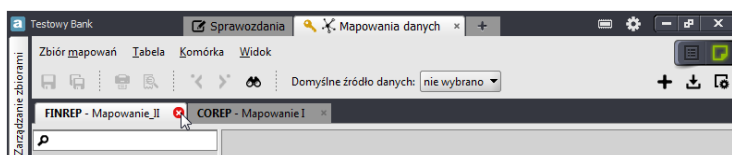
- **Menu „Zbiór mapowań”** → **Zamknij** ,

- a następnie jedną z opcji:

Zamknij aktywny zbiór
 Zamknij pozostałe zbiory
 Zamknij wszystkie zbiory


lub

- kliknąć na:  wybranej zakładki zbioru mapowań (po najechaniu kursorem, zmienia się jego zabarwienie na czerwone  – zob. Rysunek 28).



Rysunek 28. Zamykanie zbioru mapowań

Funkcja ta:

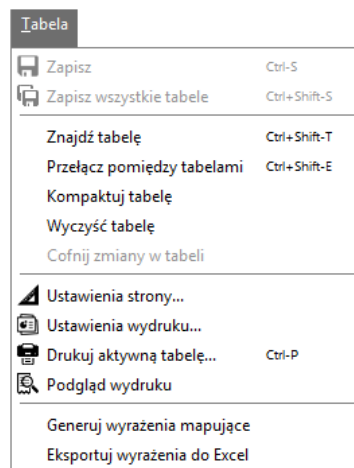
- pozwala na zamknięcie jednego bądź wielu zbiorów mapowań, poprzez wybranie odpowiedniej opcji;
- może być wybrana tylko z poziomu „Dokumenty” .

Te same funkcje są dostępne i mogą być wykorzystywane po „kliknięciu prawym przyciskiem myszki” na zakładkę jednego z otwartych zbiorów mapowań.

2.1.1.2 Tabela

Menu „Tabela” – grupuje funkcje (zob. Rysunek 29), związane z obsługą tabel aktywnego zbioru wyrażeń mapujących, do których należą:

- zapisanie zmian wprowadzonych do aktywnej i wszystkich tabel;
- szybkie odszukanie wymaganej tabeli,
- szybkie przełączanie między tabelami,
- usunięcie z aktywnej tabeli wszystkich wyrażeń mapujących,
- kompaktowanie wyrażeń w tabeli,
- ustawienie parametrów wydruku,
- drukowanie tabel z wprowadzonymi wyrażeniami mapującymi,
- generowanie wyrażeń mapujących,
- eksport zdefiniowanych wyrażeń mapujących do pliku w formacie Excel.



Rysunek 29. Menu "Tabela" w trybie „Dokumenty”

Poza dwoma funkcjami: „Generuj wyrażenia mapujące” i „Eksportuj wyrażenia do Excela”, działanie innych udostępnionych tu funkcji jest opisane w instrukcji [aSIS-Podstawy obsługi aplikacji.pdf](#) (rozdział „Tabela”) z tym, że tam funkcje te odnoszą się do tabel sprawozdawczych, a tutaj do tabel aktywnego zbioru wyrażeń mapujących.

2.1.1.2.1 Generuj wyrażenia mapujące

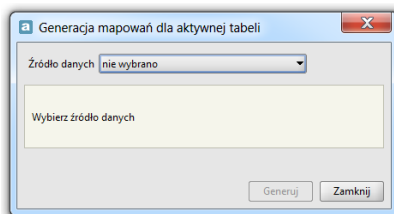
Aby uruchomić funkcję „Generuj wyrażenia mapujące”, należy wybrać:

- **Menu „Tabela”** → Generuj wyrażenia mapujące

Funkcja ta:

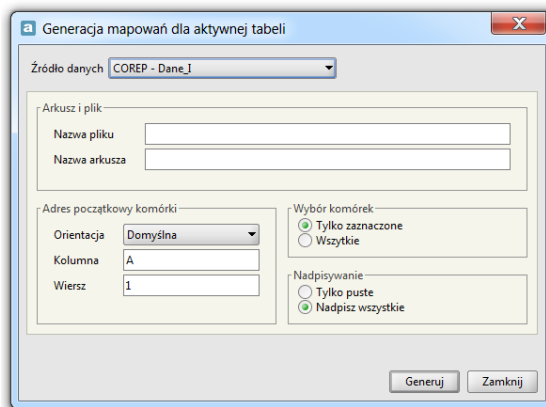
- pozwala bez konieczności ręcznego definiowania wyrażeń mapujących, na ich automatyczne wygenerowanie przez samą aplikację;
- może być wykorzystana tylko przy komórkowych źródłach danych,
- zawsze dotyczy jednej, otwartej tabeli.

Po jej uruchomieniu otwiera się okno (zob. Rysunek 30):



Rysunek 30. Wybór źródła danych przed generowaniem wyrażeń mapujących


Konieczne jest wskazanie komórkowego źródła danych w oparciu, o które system ma dokonać generowania wyrażeń mapujących, gdyż tylko wtedy funkcja się uaktywni (zob. Rysunek 31).

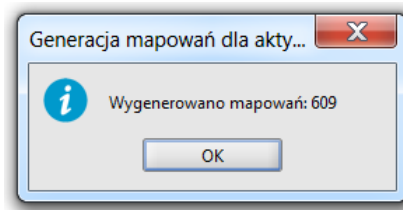


Rysunek 31. Generacja mapowań dla aktywnej tabeli

Przed właściwym uruchomieniem generowania wyrażeń mapujących konieczne jest jeszcze bliższe sprecyzowanie warunków, które powinny być spełnione, aby wygenerowane wyrażenia spełniały nasze oczekiwania, czyli:

- podanie nazwy pliku/arkusza,
- określenie adresu początkowego komórki, od której ma nastąpić definiowanie wyrażeń mapujących;
- wybór komórek, dla których mają być wygenerowane mapowania (tylko zaznaczone / wszystkie);
- określenie sposobu zapisywania przygotowanych mapowań (tylko puste / nadpisz wszystkie).

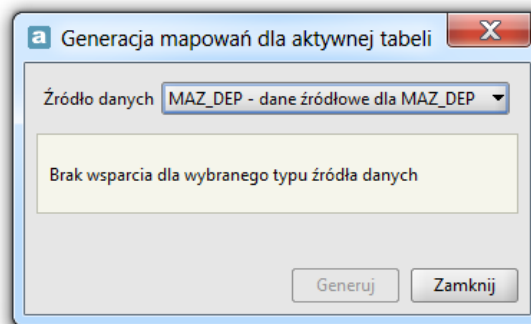
Akceptacja wprowadzonych wyborów poprzez użycie przycisku  spowoduje wygenerowanie i wprowadzenie do poszczególnych (wcześniej wybranych) komórek stosownych wyrażeń mapujących, o czym użytkownik jest informowany komunikatem (zob. Rysunek 32).



Rysunek 32. Informacja o wygenerowaniu wyrażeń mapujących

Wybranie przycisku , kończy działanie funkcji.

W przypadku wskazania źródła danych innego niż 'Excel komórkowy', funkcja nie będzie mogła się wykonać i zaprezentowany zostanie komunikat (zob. Rysunek 33).



Rysunek 33. Wybór innego niż komórkowe źródło danych, przy wykonywaniu funkcji generowania wyrażeń mapujących

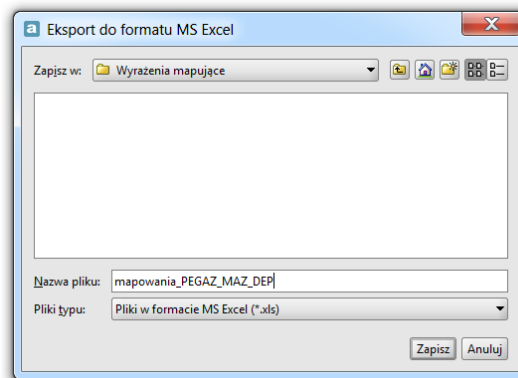
2.1.1.2.2 Eksportuj wyrażenia do Excela

Aby uruchomić funkcję „Eksportuj wyrażenia do Excela”, należy wybrać:

- **Menu „Tabela”** → Eksportuj wyrażenia do Excel

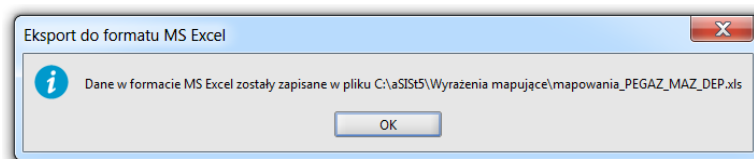
Funkcja ta:

- pozwala na wyeksportowanie w formacie Excel, zdefiniowanych wyrażeń mapujących;
- zawsze dotyczy wyrażeń mapujących zdefiniowanych z otwartej tabeli aktywnego zbioru mapowań,
- dla prawidłowego wykonania wymaga podania (zob. Rysunek 34):
 - ścieżki do zapisu,
 - nazwy pliku.



Rysunek 34. Eksport wyrażeń mapujących do Excela

O prawidłowym zakończeniu funkcji użytkownik jest informowany komunikatem (zob. Rysunek 35):

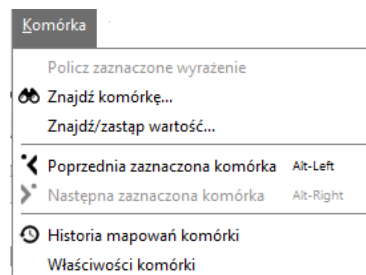


Rysunek 35. Informacja o zapisaniu pliku w formacie Excel

2.1.1.3 Komórka

Menu „Komórka” – grupuje funkcje (zob. Rysunek 36), związane z obsługą pojedynczych komórek w tabelach, czyli:

- policz zaznaczone wyrażenie,
- znajdź komórkę,
- znajdź/zastąp wartość,
- przejdź do poprzedniej/następnej zaznaczonej komórki,
- historia mapowań komórki,
- właściwości komórki.



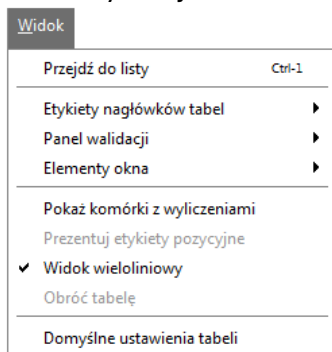
Rysunek 36. Menu "Komórka" w trybie „Dokumenty”

Działanie tych funkcji jest takie jak funkcji opisanych w [aSIS-Podstawy obsługi aplikacji.pdf](#) (zob. rozdział „Komórka”) z tym, że tam odnoszą się one do komórek w aktywnym sprawozdaniu, a tutaj dotyczą zbioru wyrażen mapujących.

2.1.1.4 Widok

Menu „Widok” grupuje (zob. Rysunek 37) funkcje, związane z ilością i sposobem prezentacji danych na ekranie monitora, czyli funkcje pozwalające na:

- przejście do listy zbiorów mapowań istniejących w bazie aSIS,
- dokonywanie zmian w sposobie prezentacji opisów tabel,
- włączanie/ wyłączenie różnych informacji wyświetlanych na ekranie: listy błędów, szczegółów zdefiniowanych wyrażen, szczegółów XBRL czy wyników wyszukiwania;
- wprowadzanie komentarzy do komórek,
- wprowadzanie zmian w ilości informacji prezentowanych na ekranie, czyli wyłączenie/włączenie: listy formularzy, panelu edycyjnego, panelu narzędzi, panelu statusu, trybu pełnoekranowego;
- prezentację komórek z wyliczeniami,
- zawijanie długich tekstów,
- obracanie tabel listowych,
- przywracające domyślnych ustawień aktywnej tabeli.



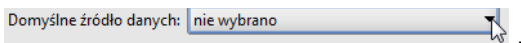
Rysunek 37. Menu "Widok" w trybie „Dokumenty”

Działanie tych funkcji jest takie jak funkcji opisanych w [aSIS-Podstawy obsługi aplikacji.pdf](#) (zob. rozdział „Widok”) z tym, że tam odnoszą się one do komórek w aktywnym sprawozdaniu, a tutaj dotyczą zbioru wyrażen mapujących.

2.1.1.5 Domyślne źródło danych

Aby zdefiniować „Domyślne źródło danych”, należy:

- rozwinąć listę dostępnych (zdefiniowanych) źródeł danych, czyli:



- wybrać jedno z wcześniej zdefiniowanych źródeł danych jako źródło domyślne.

Funkcja ta:

- może być wykorzystana tylko po wcześniejszym zdefiniowaniu wymaganych źródeł danych,
- pozwala na uproszczony sposób budowania wyrażeń mapujących.

Przykładowo:

Zdefiniowano wyrażenie, w którym w jawny sposób podano identyfikator źródła danych (**Źródło_1**):


Źródło_1: { sum(\$wplaty, \$nazwisko = Kowalski) }

Jeśli wymagane źródło danych zostanie wybrane jako domyślne, wówczas wyrażenie może mieć postać:

{ sum(\$wplaty, \$nazwisko = Kowalski) }

a symbol źródła danych zostanie włączony do wyrażenia automatycznie.

2.1.2 Tryb lista

Tryb „Lista”, dla którego przełącznik widoku dokumentów ma postać  to tryb, w którym:

- prezentowana jest lista wszystkich utworzonych zbiorów wyrażeń mapujących,
- możliwe do wykonania funkcje (zob. Rysunek 38) pogrupowane zostały w menu:
 - Zbiór mapowań,
 - Widok.

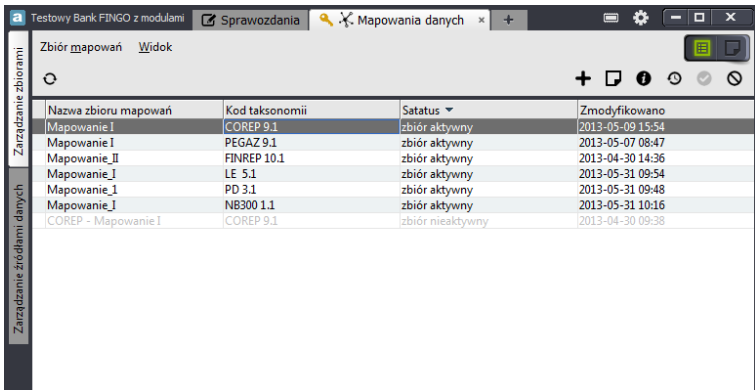
Przejście z trybu „Dokumenty” do trybu „Lista zbiorów mapowań” możliwe jest, po wybraniu:

- **Menu „Zbiór mapowań”** → Zarządzanie zbiorami

lub

- jednego ze skrótów klawiszowych: Ctrl-O lub Ctrl-I
bądź

- po przełączeniu „Przełącznika widoku dokumentów”, czy  .



Nazwa zbioru mapowań	Kod taksonomii	Status	Zmodyfikowano
Mapowanie_I	COREP 9.1	zbior aktywny	2013-05-09 15:54
Mapowanie_I	PEGAZ 9.1	zbior aktywny	2013-05-07 08:47
Mapowanie_II	FINREP 10.1	zbior aktywny	2013-04-30 14:36
Mapowanie_I	LE 5.1	zbior aktywny	2013-05-31 09:54
Mapowanie_I	PD 3.1	zbior aktywny	2013-05-31 09:48
Mapowanie_I	NE300 1.1	zbior aktywny	2013-05-31 10:16
COREP - IMapowanie I	COREP 9.1	zbior nieaktywny	2013-04-30 09:38

Rysunek 38. Zarządzanie zbiorami mapowań – lista dostępnych zbiorów wyrażeń mapujących

Na wyświetlonej liście, w poszczególnych kolumnach wyświetlane są:

- nazwa zbioru mapowań,
- kod taksonomii, którego dotyczy dany zbiór;
- status zbioru mapowań (zbiory nieaktywne – prezentowane są jaśniejszą czcionką),
- data ostatniej modyfikacji zbioru.

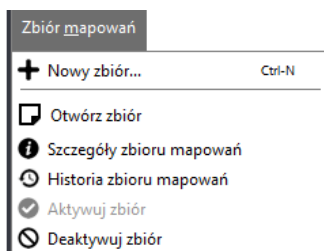
2.1.2.1 Zbiór mapowań

Menu „Zbiór mapowań” (zob. Rysunek 39), grupuje podstawowe funkcje związane z zarządzaniem zbiorami mapowań, czyli:

- tworzenie nowych zbiorów wyrażeń mapujących,
- otwieranie wcześniej utworzonych zbiorów mapowań,
- przeglądanie szczegółów zbiorów mapowań,
- zmiana aktywności zbiorów mapowań, czyli deaktywowanie/aktywowanie zbiorów mapowań.

Z wyjątkiem funkcji „Nowy zbiór”, wszystkie pozostałe funkcje:



- dotyczą konkretnego, wybranego z prezentowanej listy zbioru mapowań;
- uaktywniają się, po podświetleniu wymaganego zbioru.



Rysunek 39. Funkcje menu „Zbiór mapowań” – tryb „Lista zbiorów mapowań”

2.1.2.1.1 Nowy zbiór wyrażeń mapujących

Aby utworzyć „Nowy zbiór wyrażeń mapujących”, należy wybrać:

- Menu „Zbiór mapowań” →  Nowy zbiór...
- lub
- Ikonę: 
- bądź
- skrót klawiszowy: Ctrl-N.

Funkcja ta:



- dostępna jest w obu trybach pracy, czyli:
 - w trybie „Dokumenty” -  ,
 - w trybie „Lista” -  ,
- została omówiona w rozdziale [Nowy zbiór wyrażeń mapujących](#).

2.1.2.1.2 Otwórz zbiór wyrażeń mapujących



Aby otworzyć, wcześniej utworzony zbiór mapowań, należy:

w trybie „Lista zbiorów mapowań”  , wykorzystać funkcję „Otwórz zbiór mapowań” poprzez:

- zaznaczenie (podświetlenie) na liście zbioru, który ma zostać otwarty, a następnie wybranie:

- Menu „Zbiór mapowań” →  Otwórz zbiór
- lub
- ikony 
- bądź
- dwukrotne kliknięcie lewym przyciskiem myszki na wybrany zbiór.

Funkcja ta:

- dostępna jest tylko w trybie „Lista” -  ,
- uaktywnia się dopiero po zaznaczeniu (podświetleniu) na liście zbioru, który ma zostać otwarty;
- otwiera wybrany zbiór i powoduje automatyczne przejście aplikacji do trybu „Dokumenty”  .

Wszystkie wcześniej otwarte zbiory nadal pozostaną otwarte i widoczne w trybie „Dokumenty”, ale nie będą one aktywne a ich zakładki będą prezentowane na ciemniejszym tle.

2.1.2.1.3 Szczegóły zbioru mapowań

Aby uruchomić funkcję „Szczegóły zbioru mapowań”, należy wybrać:

- **Menu „Zbiór mapowań”** →  Szczegóły zbioru mapowań
- lub
- Ikonę: .

Funkcja ta:



- dostarcza informacji o ustawieniach zbioru wyrażeń mapujących (zob. Rysunek 10), jednak tutaj nie ma możliwości wprowadzania żadnych zmian (zob. rozdział [Ustawienia zbioru wyrażeń mapujących](#));
- uaktywnia się dopiero po zaznaczeniu (podświetleniu) na liście zbioru, którego szczegóły chcemy przeglądać.

2.1.2.1.4 Historia zbioru mapowań

Aby uruchomić funkcję „Historia zbioru mapowań”, należy wybrać:

- **Menu „Zbiór mapowań”** →  Historia zbioru mapowań
- lub
- Ikonę: .

Funkcja ta (zob. Rysunek 14), przedstawia:


- historię zmian wprowadzonych do aktywnego zbioru mapowań,
- dostępna jest w obu trybach, czyli „Lista”  i „Dokument” .
- w trybie „Lista” uaktywnia się dopiero po zaznaczeniu (podświetleniu) na liście zbioru, którego historię chcemy przejrzeć.

2.1.2.1.5 Aktywuj zbiór

Aby uruchomić funkcję „Aktywuj zbiór mapowań”, należy wybrać:

- **Menu „Zbiór mapowań”** →  Aktywuj zbiór
- lub
- Ikonę: .

Funkcja ta:

- dostępna jest tylko w trybie „Lista” -  ,
- uaktywnia się dopiero po zaznaczeniu (podświetleniu) na liście zbioru, który posiada status „nieaktywny”, czyli wcześniej został deaktywowany;
- zmienia status zbioru na „aktywny”.

Zbiór mapowań o statusie „aktywny”:


- może być otwierany i edytowany;
- sformułowane w nim wyrażenia mapujące mogą zostać wyeksportowane (zob. rozdział [Eksportuj wyrażenia mapujące](#)),
- można do niego zaimportować inne wyrażenia mapujące (zob. rozdział [Importuj wyrażenia mapujące](#)),
- może być wykorzystywany do zasilenia sprawozdań danymi zewnętrznymi (zob. rozdział [Zasilenie według mapowań](#)).

2.1.2.1.6 Deaktywuj zbiór

Aby uruchomić funkcję „Deaktywuj zbiór mapowań”, należy wybrać:

- Menu „Zbiór mapowań” →  Deaktywuj zbiór
- lub
- Ikonę:  .

Funkcja ta:

- dostępna jest tylko w trybie „Lista” -  ,
- uaktywnia się dopiero po zaznaczeniu (podświetleniu) na liście zbioru, który posiada status „aktywny”;
- zmienia status zbioru na „nieaktywny”.

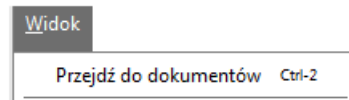
Zbiór mapowań o statusie „nieaktywny”:

- może być otwierany, ale tylko do podglądu;
- nie może być edytowany,
- sformułowane w nim wyrażenia mapujące mogą zostać wyeksportowane, ale
- nie można do niego zaimportować innych wyrażen mapujących,
- nie jest widoczny przez funkcję „Zasilenia wg mapowań” i dlatego nie może być wykorzystywany do zasilenia sprawozdań danymi zewnętrznymi;
- może zostać w dowolnym momencie aktywowany.

2.1.2.2 Widok

Menu „Widok” (zob. Rysunek 40), udostępnia funkcję związaną z:

- przejściem a do pracy w trybie „dokumenty”.



Rysunek 40. Menu "Widok" w trybie „Lista zbiorów mapujących”

2.1.2.2.1 Przejście do trybu „Dokumenty”

Przejście z trybu „Lista zbiorów mapowań” do trybu „Dokumenty” możliwe jest, po wybraniu:

- **Menu „Widok”** → *Przejdź do dokumentów*

lub

- Skrótu klawiszowego: Ctrl-2

bądź

- po przełączeniu „Przełącznika widoku dokumentów”, czy...  .

2.2 Zarządzanie źródłami danych

Uaktywnienie modułu „Mapowanie danych” i wybór bocznej zakładki „Zarządzanie źródłami danych” (zob. Rysunek 41):

- dostarcza informacji o zdefiniowanych źródłach danych, poprzez:
 - prezentację listy zdefiniowanych źródeł,
- umożliwia definiowanie nowych i zarządzanie istniejącymi źródłami danych, poprzez:
 - wprowadzanie zmian do istniejących źródeł,
 - eksportowanie i importowanie źródeł danych,
 - przeglądanie historii zmian wprowadzonych do źródeł danych,
- szybkie przeliczenie zdefiniowanych wyrażeń mapujących, w oparciu o wskazane źródło danych.

Typ źródła	Identyfikator	Długa nazwa	Połączenie
Excel kolumnowy	MAZ_GOT	dane źródłowe dla MAZ_GOT	C:\JENKINS\aSIST5-2_16-04-2013\mapowania\zródl...
Excel kolumnowy	MAZ_DEP	dane źródłowe dla MAZ_DEP	C:\JENKINS\aSIST5-2_16-04-2013\mapowania\zródl...
Excel komórkowy	w	w	C:\Users\Jadwiga\Desktop\test.xls
Excel komórkowy	COREP	Dane_I	C:\Users\Jadwiga\Desktop\Źródła danych dla map...

Rysunek 41. Zarządzanie źródłami danych

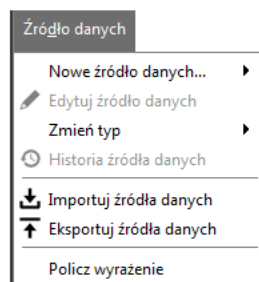
Wszystkie funkcje zostały tu udostępnione:

- w menu „Źródło danych” oraz
- w formie ikon funkcyjnych.

2.2.1 Źródło danych

Menu „Źródło danych” (zob. Rysunek 42), grupuje funkcje związane z:

- zarządzaniem źródłami danych,
- szybkim przeliczeniem dowolnego wyrażenia mapującego, wykorzystującego dostępne źródła danych.



Rysunek 42. Menu „Źródło danych” w Zarządzaniu źródłami danych

Źródła danych są:


- globalne dla całej aplikacji,
- współdzielone przez wykorzystujące je wyrażenia mapujące.

2.2.1.1 Nowe źródło danych

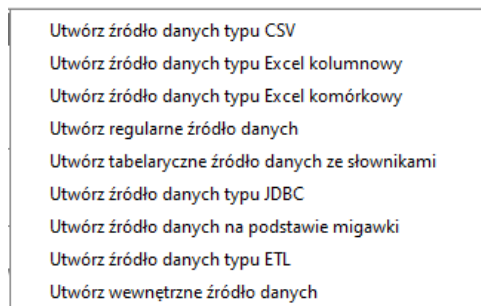
Aby zdefiniować „Nowe źródło danych”, należy w zakładce „Zarządzanie źródłami danych”, wybrać:

- Menu „Źródło danych” → Nowe źródło danych...

lub

- Ikonę: ,

a następnie jedną z opcji:



Funkcja ta pozwala na zdefiniowanie nowego źródła danych:

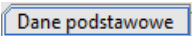
- o typie zgodnym z dokonanym wyborem, przy uruchamianiu tej funkcji;
- nie powiązanego z żadną taksonomią ani sprawozdaniem – mającego charakter globalny dla całej aplikacji,
- współdzielonego przez wykorzystujące je wyrażenia mapujące.

Po jej uruchomieniu wyświetla się jeden z ekranów (zob. Rysunek 43, Rysunek 44, Rysunek 45, Rysunek 46, Rysunek 47, Rysunek 48, Rysunek 49, Rysunek 50, Rysunek 51).

Dla różnych typów źródeł danych, wymagana jest różna ilość danych do zdefiniowania, pogrupowanych w zakładkach:

- dane podstawowe,
- parametry,
- kolumny wirtualne,
- formaty,
- słowniki.

2.2.1.1.1 Dane podstawowe nowego źródła danych

W zakładce  (zob. Rysunek 43, Rysunek 44, Rysunek 45, Rysunek 46, Rysunek 47, Rysunek 48, Rysunek 49, Rysunek 50, Rysunek 51), wymagane jest podanie:

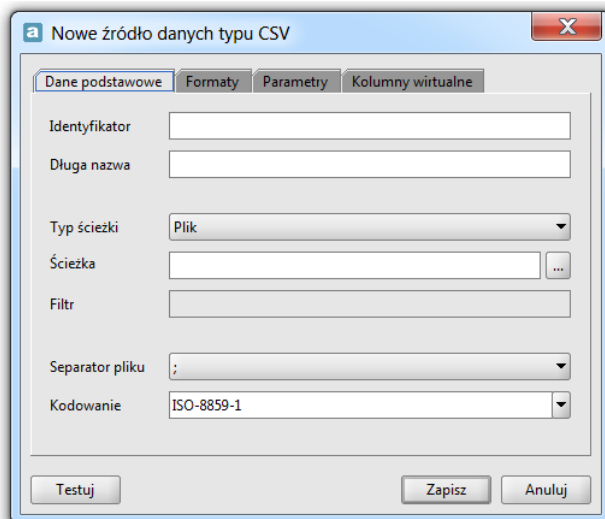
- **identyfikatora** – unikatowego kodu definiowanego źródła danych, wykorzystywanego w budowanych wyrażeniach – dla wszystkich typów źródeł danych;
- **nazwy** – pełnej nazwy definiowanego źródła danych – dla wszystkich typów źródeł danych (za wyjątkiem „Migawkowego źródła danych”, „Wewnętrznego źródła danych”),

- **nazwy migawki** – dla „Migawkowego źródła danych”,
- **identyfikatora procesu** – dla „Źródła danych typu ETL”,
- **kroku procesu** – dla „Źródła danych typu ETL”,
- **zapytania SQL** – dla „Migawkowego źródła danych”,
- **typu ścieżki danych**, prezentowanych w postaci:
 - pliku,
 - katalogu,
 - archiwum zip,możliwość wyboru dla wszystkich plikowych źródeł danych (z wyjątkiem „Tabeli ze słownikami”, „Migawkowego źródła danych”, „Wewnętrznego źródła danych” oraz „Źródła danych typu ETL”),
- **ścieżki dostępu do danych** – dla wszystkich plikowych źródeł danych (z wyjątkiem „Migawkowego źródła danych”, „Wewnętrznego źródła danych” oraz „Źródła danych typu ETL”),
filtra – umożliwiającego dodatkowe filtrowanie plików w przypadku wybrania katalogu lub archiwum ZIP dla wszystkich plikowych źródeł danych (z wyjątkiem „Wewnętrznego źródła danych” oraz „Źródła danych typu ETL”),
- **separatora danych w postaci:**
 - ;
 - |
 - <TAB>

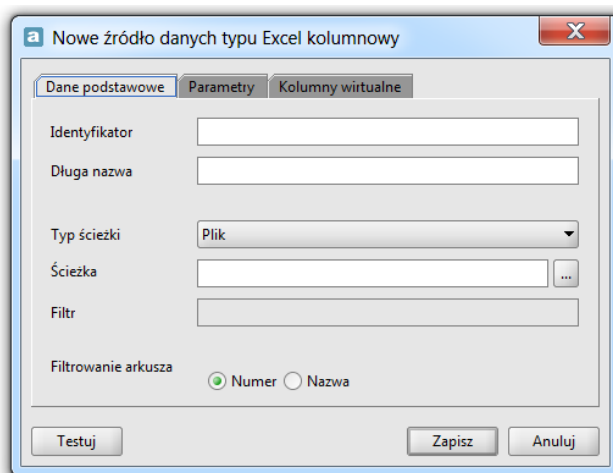
dla plików CSV (zob. Rysunek 43):

- **sposobu kodowania:**
 - ISO-8859-1 (ustawienie domyślne),
 - ISO-8859-2,
 - Windows-1250,
 - UTF-8,
 - UTF-16,
 - US-ASCII,
 - inne ,**dla źródła danych typu CSV i Regularne,**
- **numeru arkusza** bądź **nazwy arkusza** – dla plików Excel kolumnowy i Excel komórkowy (zob. Rysunek 44 i Rysunek 45);
- **typ regularnej tabeli źródła danych:**
 - regularny nagłówek,
 - regularna tabela lub
 - stałe kolumny,dla plików typu: Regularne (zob. Rysunek 46),
- **podstawowe źródło danych** – wskazujące inne istniejące źródło danych dla danych typu „Tabela ze słownikami” (zob. Rysunek 47),

- **zbioru taksonomii** dla „Wewnętrznego źródła danych” (zob. Rysunek 50).



Rysunek 43. Definiowanie nowego źródła danych typu CSV



Rysunek 44. Definiowanie nowego źródła danych typu Excel kolumnowy

Nowe źródło danych typu Excel komórkowy

Dane podstawowe Parametry Kolumny wirtualne

Identyfikator

Nazwa

Typ ścieżki Plik

Ścieżka

Filtr

Filtrowanie arkusza Numer Nazwa

Arkusz

przetwarzanie strumieniowe (tylko *.xlsx)

Przetwarzanie strumieniowe wymaga mniej pamięci, jednak w jego trakcie nie są wyliczane wartości komórek formułowych - brane pod uwagę są wyłącznie wyniki formuł zapisane w pamięci podręcznej skoroszytu. W celu wyliczenia wartości komórek formułowych i zapisania ich w pamięci podręcznej skoroszytu należy użyć zewnętrznego narzędzia (np. Microsoft Excel).

Testuj Zapisz Anuluj

Rysunek 45. Definiowanie nowego źródła danych Excel komórkowy

Nowe źródło danych typu Regularne

Dane podstawowe Tabela Formaty Parametry Kolumny wirtualne

Identyfikator

Długa nazwa

Typ ścieżki Plik

Ścieżka

Filtr

Typ Regularny nagłówek

Kodowanie ISO-8859-1

Testuj Zapisz Anuluj

Rysunek 46. Definiowanie nowego źródła danych typu Regularne

The dialog box is titled "Nowe źródło danych typu Tabela ze słownikami". It has four tabs: "Dane podstawowe", "Słowniki", "Formaty", and "Parametry". The "Dane podstawowe" tab is active. It contains three input fields: "Identyfikator" (empty), "Długa nazwa" (empty), and "Podstawowe źródło danych" (a dropdown menu with "nie wybrano" selected). At the bottom right, there are "Zapisz" and "Anuluj" buttons.

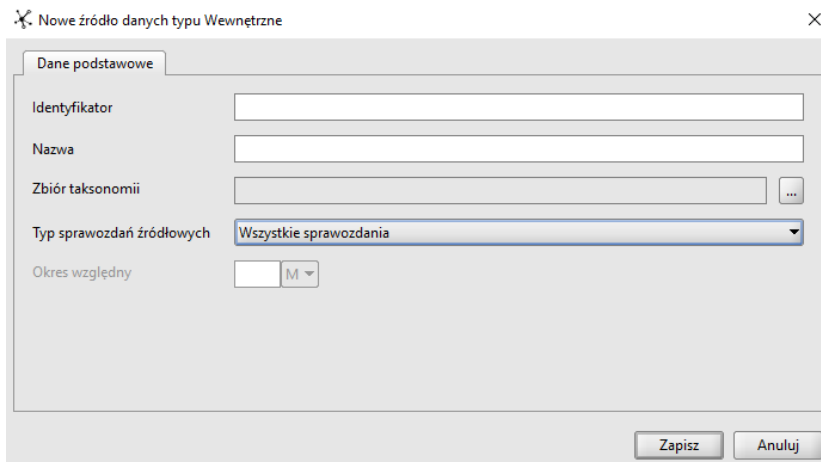
Rysunek 47. Definiowanie nowego źródła danych typu Tabela ze słownikami

The dialog box is titled "Nowe źródło danych typu JDBC". It has three tabs: "Dane podstawowe", "Parametry", and "Kolumny wirtualne". The "Dane podstawowe" tab is active. It contains several input fields: "Identyfikator" (empty), "Długa nazwa" (empty), "Typ" (dropdown menu with "ORACLE" selected), "Login" (empty), "Hasło" (empty), "Połączenie" (empty), and "Zapytanie" (empty text area). There is a green checkmark icon at the bottom right of the input area. At the bottom of the dialog, there are "Testuj", "Zapisz", and "Anuluj" buttons.

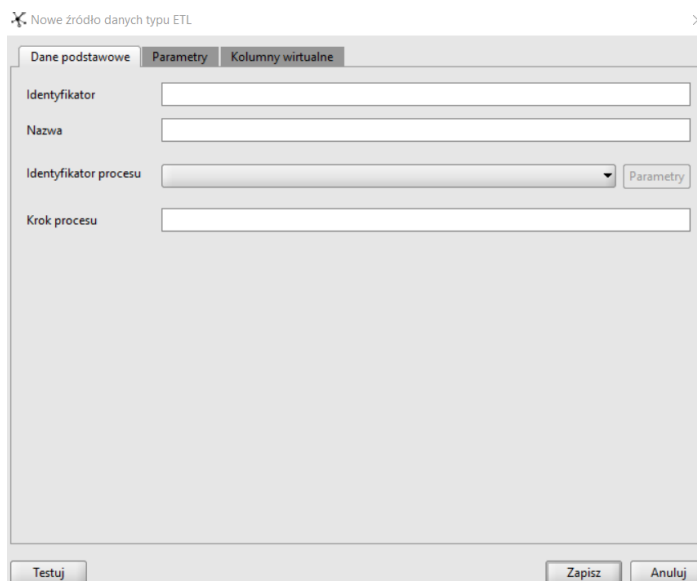
Rysunek 48. Definiowanie nowego źródła danych typu JDBC

The dialog box is titled "Nowe źródło danych typu Tabela z migawki". It has three tabs: "Dane podstawowe", "Parametry", and "Kolumny wirtualne". The "Dane podstawowe" tab is active. It contains four input fields: "Identyfikator" (empty), "Nazwa" (empty), "Nazwa migawki" (dropdown menu with "MIGAWKA_1" selected), and "Zapytanie" (empty text area). There are checkmark and edit icons at the bottom right of the input area. At the bottom of the dialog, there are "Testuj", "Zapisz", and "Anuluj" buttons.

Rysunek 49. Definiowanie nowego źródła danych typu „Migawkowe źródło danych”




Rysunek 50. Definiowanie wewnętrznego źródła danych



Rysunek 51. Definiowanie nowego źródła danych typu ETL

Po wprowadzeniu wymaganych danych, dla wszystkich źródeł danych, z wyjątkiem „**Tabeli ze słownikami**”, „**Wewnętrznego źródła danych**”, możliwe jest sprawdzenie poprawności zdefiniowanych ustawień i wyświetlenie fragmentu testowych danych.

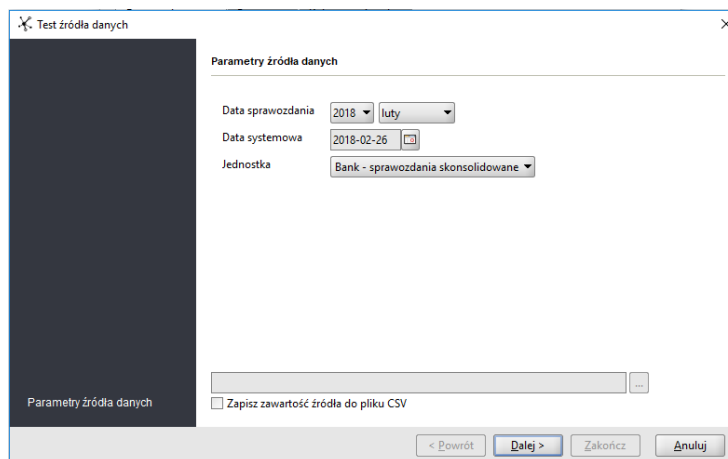
Dokonanie weryfikacji poprawności ustawień jest możliwe podczas procesu testowania źródła danych składającego się z 4 kroków.

Aby uruchomić wspomnianą funkcję należy wcisnąć przycisk  (zob. Rysunek 43, Rysunek 44, Rysunek 45, Rysunek 46, Rysunek 48, Rysunek 49, Rysunek 51), a następnie:

Krok 1, to: Wskazanie parametrów źródła danych (zob. Rysunek 52), czyli:

- daty sprawozdania, będącej okresem sprawozdawczym;
- daty systemowej, zdefiniowanej w źródle za pomocą funkcji „getSystemDate()”;

- o jednostki oraz opcjonalnie,
- **zapisaniu zawartości źródła do pliku CSV**, poprzez wybranie przycisku: , a następnie za pomocą: wybranie pliku do zapisu interpretacji źródła danych.



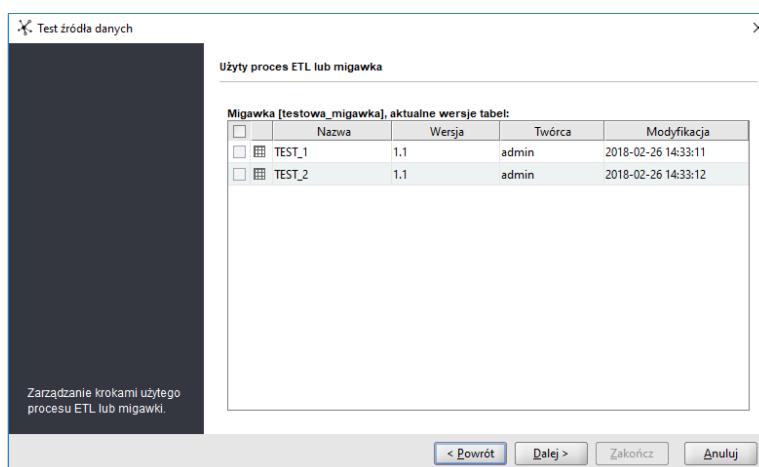
Rysunek 52. Parametryzowanie źródła danych typu ETL

Aby przejść do kroku 2, należy wybrać przycis .

Krok 2, to: Prezentacja informacji na temat użytego procesu ETL lub migawki. Krok ten występuje tylko przy testowaniu źródła danych typu ETL lub źródła danych typu migawkowego.

Okno w zależności od typu źródła danych może mieć postać:

- dla „**Migawkowego źródła danych**” (zob. Rysunek 53):



Rysunek 53. Okno prezentujące proces ETL lub migawkę użytą do testowania źródła migawkowego

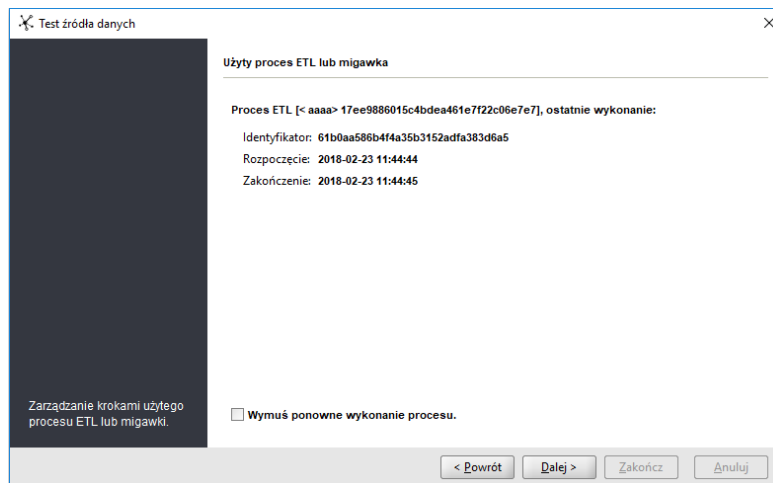
W kroku drugim podczas testowania źródła danych typu migawkowego, prezentowane jest okno (zob. Rysunek 53) w którym widoczne są:

- o wszystkie tabele utworzone w ramach danego źródła,

- o informacje na temat: nazwy migawki, jej wersji, twórcy oraz daty ostatniej modyfikacji tabel.

Użytkownik ma możliwość wybrać table do testu, poprzez:

- o zaznaczenie okienka znajdującego się przed nazwą tabeli.
- dla „**Źródła danych typu ETL**” (zob. Rysunek 54):



Rysunek 54. Okno prezentujące proces ETL lub migawkę użytą do testowania źródła ETL

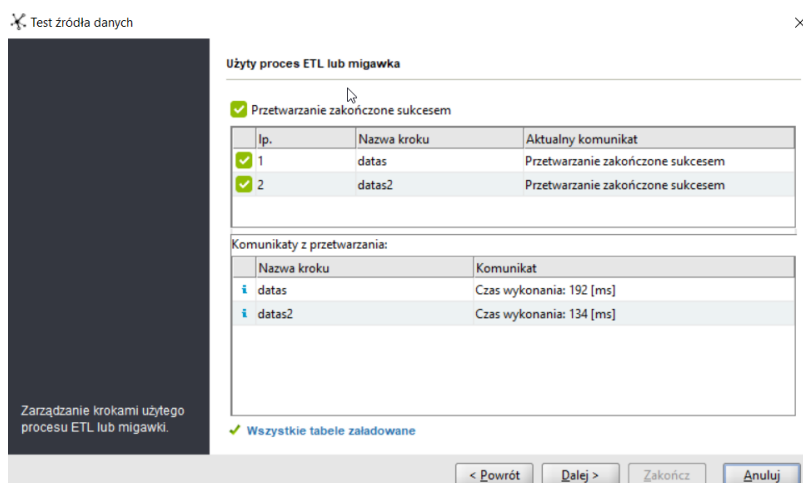
W przypadku źródła danych typu ETL w kroku drugim prezentowane są informacje na temat

- o identyfikatora,
- o czasu rozpoczęcia i zakończenia ostatniego wykonanego testu dla tego procesu.

Użytkownik ma możliwość:

- o wybierając przycisk podejrzeć wynik ostatniego wykonania lub
- o wymusić ponowne wykonanie tego testu zaznaczając przycisk , znajdujący się w dolnej części okna.

Krok 3, to: Przetwarzanie procesu ETL lub migawki. Krok ten podobnie jak krok 2 występuje tylko przy testowaniu źródła danych typu ETL lub źródła danych typu migawki. Okno bez względu na typ źródła danych przyjmuje taką samą postać jak poniżej (zob. Rysunek 55):

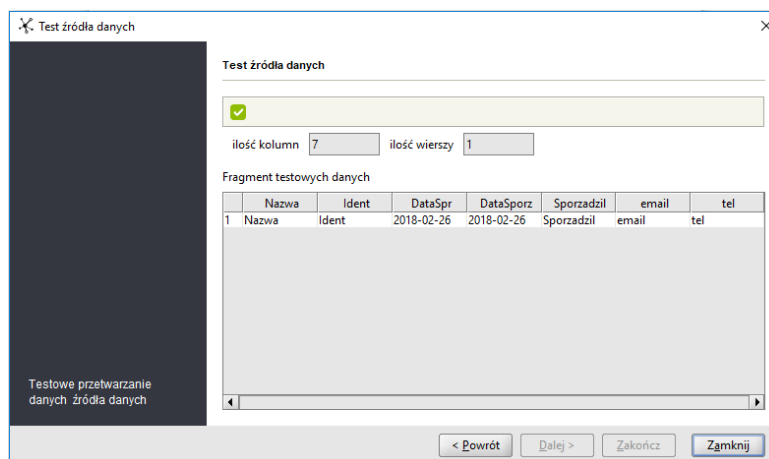


Rysunek 55. Przetwarzanie procesu ETL lub migawki

W kroku trzecim następuje dla

- **źródła danych typu migawka** wywołanie zapytania SQL bqdź
- **źródła danych typu ETL** wywołanie procesu ETL na serwerze.

Krok 4, to: Prezentacja wyniku testu źródła danych (zob. Rysunek 56):



Rysunek 56. Test źródła danych ETL

Na liście widoczny jest fragment testowanych danych. Aplikacja wyświetla maksymalnie 10 pierwszych wierszy.

Powyżej fragmentu danych, widoczna jest informacja o ilości wierszy i kolumn testowanego źródła danych.

W przypadku wszystkich rodzajów testowanych źródeł, okno to ma taki sam wygląd, różni się jedynie wyświetlanymi danymi.

Tabelaryczne źródła danych.

Wszystkie wspierane źródła danych wymagają, aby definicja źródła opisywała, jak z danych źródłowych uzyskać poprawną tabelę.

Prawidłowo przygotowane tabelaryczne źródło danych to tabela posiadająca:

- nagłówki z unikatowymi nazwami kolumn,
- wiersze z danymi posiadające tą samą liczbę kolumn.

Nazwy kolumn muszą być zgodne z formatem:

[a-zA-Z][a-zA-Z0-9_]*

Oznacza to, że wszystkie znaki w nazwach kolumn niepasujące do tego wzorca (również białe znaki) będą zamieniane na znak: '_'.

Białe znaki z początku i końca nazwy kolumny są usuwane. Na przykład nazwa:

' **saldo pocz.** '

w czasie wczytywania źródła zostanie zamieniona na:

'**saldo_pocz_**'

W zależności od typu źródła danych, możliwe jest pojawienie się dodatkowych kolumn wypełnianych przez system. Takie kolumny nazywane są systemowymi, a ich nazwa zaczyna się od prefiksu 'sys'. Odwołanie do kolumny systemowej 'value' wygląda tak:

\$sys.value

Źródła danych typu CSV, Excel kolumnowy, Excel komórkowy i regularne są plikowymi źródłami tabelarycznymi. Oznacza to, że dane pobierane są z tabeli zdefiniowanej w pliku (lub plikach) odpowiadających typowi źródła danych.

Ścieżka wskazuje zatem plik, katalog lub archiwum ZIP z plikami w odpowiednim formacie.

W przypadku wskazania katalogu lub archiwum ZIP zamiast jednego pliku możliwe jest wczytanie wielu plików w tym samym formacie.

Oznacza to, że wiele plików zostanie połączonych i traktowanych jako jedna tabela z wierszami ze wszystkich plików.

Taka operacja wymaga, aby wszystkie wskazane pliki posiadały tabelę o dokładnie takim samym nagłówku.

Aby ograniczyć nazwy plików wczytywanych w ramach katalogu lub archiwum ZIP należy zdefiniować **filtr**.

Filtry korzystają z prostego mechanizmu wyrażeń regularnych:

- * oznacza dowolny napis,
- ? oznacza jeden znak,
- [<znaki>], oznacza jeden znak ze zbioru <znaki>.

Na przykład:

[abc]*.csv

Oznacza wszystkie pliki zaczynające się na 'a', 'b' lub 'c' i kończące '.csv'.

Domyślnie filtr jest pusty, czyli tożsamy z filtrem '*'.

Pozostałe atrybuty definicji źródła danych zależą od jego typu.

Typ CSV, (ang. Comma Separated Values - wartości rozdzielone przecinkiem) to źródło plikowe, w którym:

- ścieżka wskazuje plik lub pliki tekstowe zawierające wartości oddzielone wskazanym separatorem,
- nazwy kolumn znajdują się w pierwszym wierszu pliku, pozostałe wiersze zawierają dane;
- poszczególne pola oddzielone są poprzez wybrany separator (";" lub "," lub "|" lub <TAB>).
W przypadku wskazania nieprawidłowego separatora może zdarzyć się, że wszystkie pola zostaną zlepione w jedną kolumnę;
- kodowanie pliku pozwala na prawidłowe rozpoznanie znaków narodowych.

Typ 'Excel kolumnowy' i 'Excel komórkowy', to źródła, w których:

- ścieżka wskazuje plik lub pliki zgodne z formatem Excel 97-2003,
- arkusz wskazuje numer lub nazwę arkusza:
 - w przypadku wskazania numeru - wartość musi być dodatnim numerem istniejącego arkusza w pliku (arkusze numerowane są od 1),
 - w przypadku wskazania nazwy - wartość jest filtrem na nazwę arkusza co oznacza, że może wskazywać więcej niż jeden arkusz. Filtr na arkusz jest konstruowany tak samo jak filtr na nazwę pliku (opisany powyżej),
- daty występują w komórkach o typie tekstowym.

Dodatkowo w **'Excelu kolumnowym'**:

- każda kolumna ma unikatowy nagłówek, nagłówek znajduje się w pierwszym wierszu wybranego arkusza;
- tabela rozpoczyna się w pierwszej kolumnie i pierwszym wierszu wybranego arkusza (komórka A1);
- wszystkie pozostałe wiersze zawierają dane tylko w kolumnach zawierających nagłówek,
- arkusz nie zawiera innych danych poza tymi znajdującymi się w tabeli.

Dodatkowo w **'Excelu komórkowym'**:


- wskazywane są arkusze z dowolnymi danymi,
- dane układane są w tabelę o nagłówku:

sys.address	sys.column	sys.row	sys.value
A1	A	1	<wartość>


wspierany jest klasyczny język wyrażeń na tabelach oraz uproszczona notacja pozwalająca odwołać się do wartości dokładnie jednej komórki arkusza, na przykład: **@A1**

- możliwe jest przetwarzanie strumieniowe plików xlsx, czyli dużych plików źródłowych przy mniejszym zużyciu pamięci aplikacji, przy czym:
 - należy przed zasileniem wyliczyć formuły i makra, a następnie zapisać plik;
 - struktura źródeł z wykorzystaniem funkcji zasilania strumieniowego jest taka sama jak w przypadku mniejszych plików, dla których nie jest wykorzystywana ta funkcja;
 - w pliku źródłowym nie należy używać danych typu księgowego (zawierającego separator) oraz typu liczbowego zawierającego separator - należy korzystać z typu liczbowego bez separatora;
 - domyślny poziom kompresji plików, dla którego plik nie jest traktowany jako złośliwy to 0.005 (200-krotna kompresja);
 - zmiana poziomu kompresji plików odbywać się może przez dodanie do aSISt.ini pozycji wraz z odpowiednią wartością:
-Dpoi.zip.secure.file.min.inflate.ratio=

Typ „Regularne źródło danych” pozwala na wskazanie w jaki sposób w pliku tekstowym została zawarta tabela:

- ścieżka wskazuje plik lub pliki tekstowe, w których znajduje się tabela z danymi;
- kodowanie pliku pozwala na prawidłowe rozpoznanie znaków narodowych,
- typ tabeli określa zawartość zakładki  , a tym samym sposób znajdowania tabeli w pliku.

Typ „Tabelaryczne źródło danych ze słownikami”, pozwala łączyć ze sobą wiele uprzednio zdefiniowanych tabelarycznych źródeł danych:

- zawiera wskazanie do głównego źródła danych,
- tabela w tym źródle ma taką samą postać jak w podstawowym źródle danych
- zakładka  umożliwi zbudowanie słowników w oparciu o istniejące źródła danych.

Typ „Migawkowe źródło danych”, pozwala za pomocą utworzonego zapytania SQL, na łączenie ze sobą wielu różnych tabel pochodzących z migawki lub kolejnych kroków przekształceń ETL, zdefiniowanych w formie migawek w „Magazynie migawek”:

- zawiera wskazanie do migawek z „Magazynu migawek”,
- tabela w tym źródle może mieć zupełnie inną postać niż tabele w podstawowym źródle danych,
- zapytanie SQL określa sposób budowania nowego źródła danych w oparciu o zdefiniowane w formie migawek tabelaryczne dane źródłowe.

Typ „Wewnętrzne źródło danych”, pozwala na łączenie ze sobą różnych źródeł z źródłem wewnętrznym (zob. Rysunek 50):

- posiada unikatowy **identyfikator** zgodny z wyrażeniem:

[a-zA-Z][a-zA-Z0-9_]*. Identyfikator może mieć taką samą nazwę jak wewnętrzne źródło danych.

Możliwe jest stworzenie kilku źródeł danych o tej samej **nazwie**.

Istotne jest przypisanie do źródła danych wymaganego **zbioru taksonomii**. W przypadku wybrania kilku taksonomii należy pamiętać, aby:

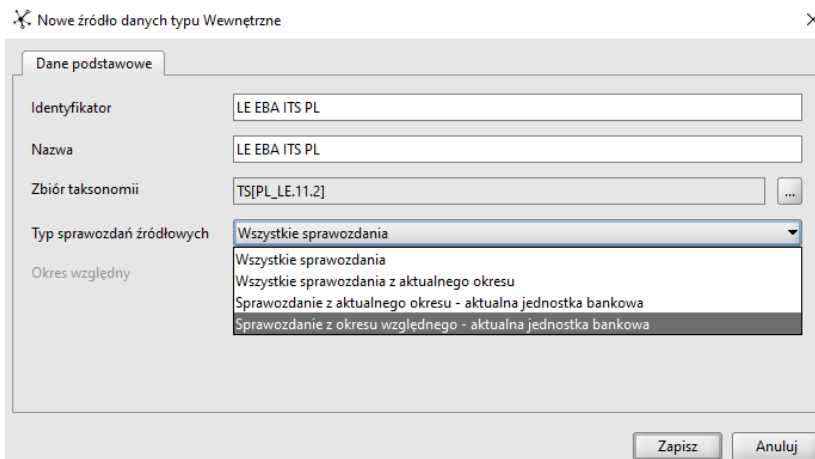
- ich typy były kompatybilne
- oraz
- aby nie posiadały rozłącznych okresów obowiązywania.

Typ danych „ETL“:

- pozwala na wykorzystanie danych będących wynikiem wykonania procesu ETL,
- każdy proces ETL składa standardowo z następujących elementów
 - wejście źródła danych (input) – plik wejściowy, przykładowo w formacie.csv;
 - kroki procesu (steps) – kroki przetwarzające plik wejściowy, np. zapytania sql pobierające dane z pliku wejściowego i przetwarzające je na potrzeby procesu;
 - wyjście – zapisanie (output) – do pliku wynikowego przykładowo w formacie .csv lub do tabeli w zewnętrznej bazie danych.

2.2.1.1.2 Wewnętrzne źródło danych – typy sprawozdań źródłowych.

Nowe źródło danych typu wewnętrznego pozwala na wybór typu sprawozdań źródłowych.

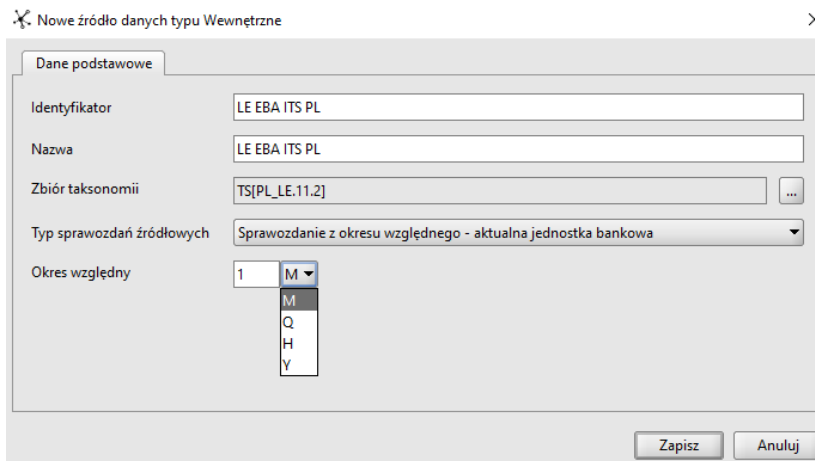


Rysunek 57. Typy sprawozdań źródłowych

Dostępne w tym polu opcje:

- **Wszystkie sprawozdania** – podczas zasilania sprawozdania jako źródło do zasilenia danymi, będą dostępne wszystkie instancje sprawozdań dla wybranego zbioru taksonomii;
- **Wszystkie sprawozdania z aktualnego okresu** – dostępne będą wszystkie sprawozdania w obrębie wybranego zbioru taksonomii, ale tylko dla okresu zgodnego z okresem, na który utworzone jest sprawozdanie;
- **Sprawozdanie z aktualnego okresu** – aktualna jednostka bankowa – dostępne będzie jedno sprawozdanie z aktualnego okresu, dla wybranego zbioru taksonomii i jednostki bankowej zgodnej ze sprawozdaniem;

- **Sprawozdanie z okresu względnego** – aktualna jednostka bankowa – dostępna będzie jedno sprawozdanie dla wybranego zbioru taksonomii i jednostki bankowej zgodnej ze sprawozdaniem, z uwzględnieniem ustawionego w definicji źródła danych okresu względnego. Można ustawiać odniesienia względem liczby:
 - M – miesięcy,
 - Q – kwartałów,
 - H – półroczy,
 - Y – lat.

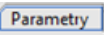


Rysunek 58. Wybór okresu względnego

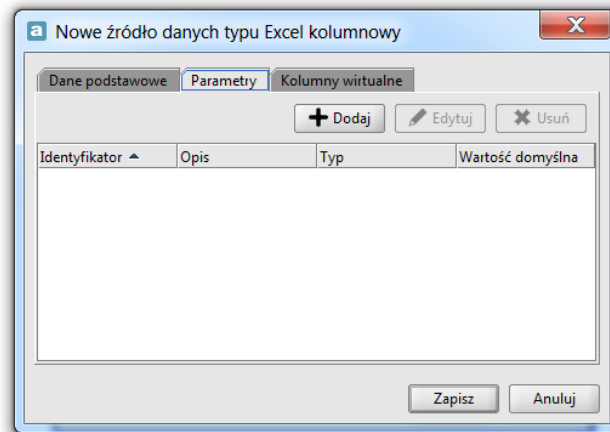
Aplikacja aSIST będzie automatycznie wyszukiwała wskazane sprawozdania dla jednostki, dla której utworzono sprawozdanie i dla okresu obliczanego jako data sprawozdania **minus wartość podana w polu**, w odniesieniu do wybranego typu okresu (miesiące/kwartałów/półroczy/lat).

*Przykład: ustawiając w polu „Okres względny” wartości **1** i **M**, będziemy w stanie zasilić sierpniowe miesięczne sprawozdanie danymi ze sprawozdania lipcowego. Dla wartości **4** i **M**, sierpniowe sprawozdanie zostanie zasilone danymi ze sprawozdania kwietniowego.*

2.2.1.1.3 Parametryzowane źródło danych

Przy wykorzystaniu zakładki  :

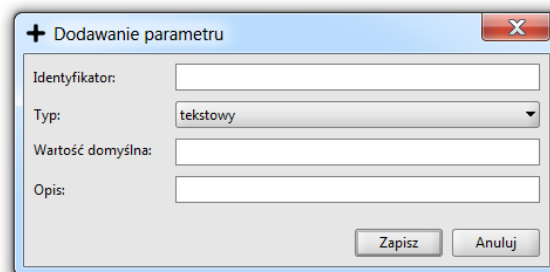
- istnieje możliwość zdefiniowania parametrów wykorzystywanych przy parametryzowanym źródle danych,
- wyświetla się ekran (zob. Rysunek 59):



Rysunek 59. Zarządzanie parametrami przy parametryzowanym źródle danych

Wybranie zakładki , pozwala na:

- utworzenie nowego parametru, wykorzystywanego przy parametryzowanym źródle danych (zob. Rysunek 60):




Rysunek 60. Definiowanie nowego parametru dla parametryzowanego źródła danych

Możliwe jest tu zdefiniowanie parametru typu:

- tekstowego,
- liczbowego,
- logicznego,
- data.

Przy zarządzaniu parametrami dla parametryzowanych źródeł danych wykorzystywane są dodatkowe klawisze funkcyjne:

 – pozwalający na edycję wcześniej zdefiniowanego parametru,

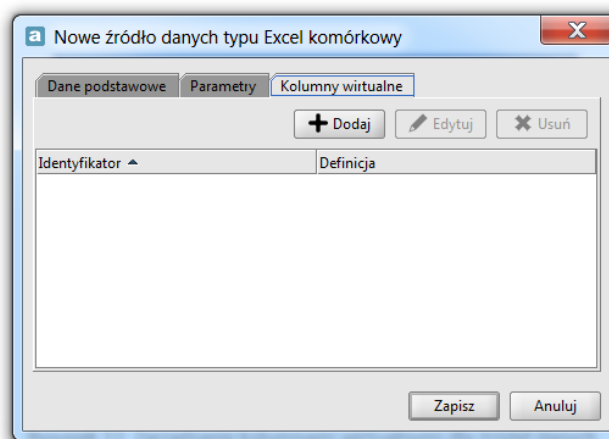
 – pozwalający na usunięcie wcześniej zdefiniowanego parametru.

Jeśli zdefiniowane tu parametry zostaną użyte przy określaniu ścieżki dostępu do źródła danych, wówczas podczas zasilania danymi, każdorazowo system odwołuje się do zdefiniowanych tu parametrów,

2.2.1.1.4 Tworzenie kolumn wirtualnych dla źródeł danych

Przy wykorzystaniu zakładki **Kolumny wirtualne** :

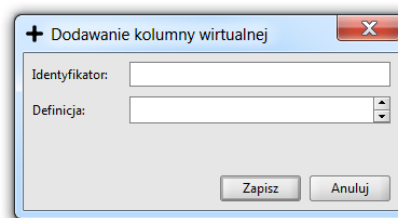
- wszystkie tabelaryczne źródła danych pozwalają na tworzenie „kolumn wirtualnych” (nie istniejących w naszym źródle danych), dla których definiowane są wyrażenia mapujące (zob. Rysunek 61):



Rysunek 61. Zarządzanie kolumnami wirtualnymi dla źródeł danych

Wybranie zakładki **+ Dodaj** , pozwala na:

- utworzenie kolumny wirtualnej,
- zdefiniowanie dla niej wyrażenia mapującego (zob. Rysunek 62):



Rysunek 62. Dodawanie kolumny wirtualnej dla źródeł danych

Kolumny wirtualne muszą posiadać unikatowe w ramach źródła identyfikatory. Nałożenie identyfikatora kolumny wirtualnej z nazwą kolumny w tabeli źródłowej powoduje błąd przy użyciu źródła danych.

Definicja kolumny wirtualnej, to wyrażenie pozwalająca na wyliczenie wartości dla każdego rekordu.

Definicja może korzystać z kolumn w źródłowej tabeli, ale nie może wykorzystywać innych kolumn wirtualnych.

Wyrażenie kolumny wirtualnej może wykorzystywać wszystkie konstrukcje języka źródła danych z wyjątkiem funkcji agregujących, co oznacza, że wartość kolumny wirtualnej dla określonego rekordu (wiersza) może zależeć jedynie od wartości innych pól tego rekordu.

\$saldo * \$kurs

Identyfikator kolumny pozwala odwoływać się do jej wartości w wyrażeniach mapujących w ten sam sposób co do pozostałych kolumn źródła danych:

sum(\$kolumna_wirtualna)

Utworzone tu kolumny wirtualne w znacznym stopniu mogą uprościć wyrażenia mapujące pozwalając uzyskać wartości pośrednie dla każdego rekordu danych źródłowych.

Przy zarządzaniu kolumnami wirtualnymi wykorzystywane są dodatkowe klawisze funkcyjne:



- pozwalający na edycję wcześniej zdefiniowanej kolumny wirtualnej,

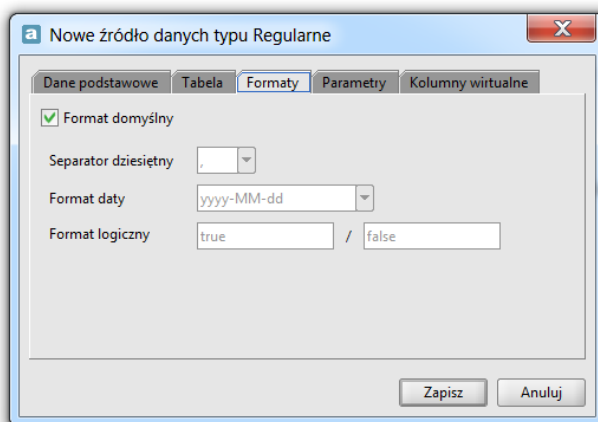


- pozwalający na usunięcie wcześniej zdefiniowanej kolumny wirtualnej.

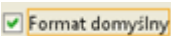
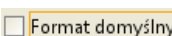
2.2.1.1.5 Formaty źródła danych typu CSV i Regularny

Przy wykorzystaniu zakładki  :

- można dokonać zmian w formacie źródła danych typu CSV, Regularne oraz tabelaryczne źródło danych ze słownikami (zob. Rysunek 63):



Rysunek 63. Formaty dla źródła danych typu Regularne

Po wyłączeniu opcji , czyli przy stanie  istnieje tu możliwość wprowadzenia zmian w zakresie formatów używanych w źródle danych:

- separatorów dziesiętnych (',' czy ';'),

- formatów daty:
 - yyyy-MM-dd,
 - yyyy.MM.dd,
 - dd-MM-yyy,
 - dd.MM.yyy,
 - innych podanych przez użytkownika,
- formatów logicznych:
'<a>|' gdzie a i b to odpowiednio wartość prawdziwa i fałszywa.

2.2.1.2 Edytuj źródło danych

Aby uruchomić funkcję „Edytuj źródło danych”, należy w zakładce „Zarządzanie źródłami danych”, wybrać:

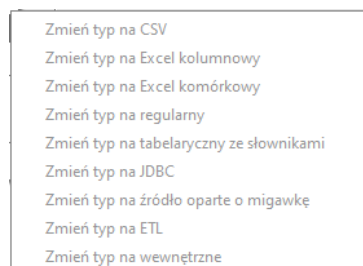
- Menu „Źródło danych” →  Edytuj źródło danych
- lub
- Ikonę: .

Funkcja ta:

- pozwala na wprowadzenie zmian do wcześniej zdefiniowanego źródła danych,
- dotyczy zawsze zaznaczonego (podświetlonego) na liście źródła danych,
- wyświetla ekran jak przy definiowaniu nowego źródła danych, jednakże teraz zawiera on informacje wcześniej zapisane, które mogą zostać zmodyfikowane;
- nie pozwala na wprowadzenie zmian w identyfikatorze źródła danych.

Aby uruchomić funkcję „Zmień typ źródła danych”, należy w zakładce „Zarządzanie źródłami danych”, wybrać:

- Menu „Źródło danych” →  Zmień typ ,
- a następnie jeden z właściwych typów:



Funkcja ta:

- pozwala dla wcześniej zdefiniowanego źródła danych (podświetlonego), na dokonanie zmiany typu tego źródła danych na właściwy;
- nie powoduje konwertowania wskazanego pliku na nowy typ,
- może być wykorzystywane, jeśli przy definiowaniu źródła danych został omyłkowo wybrany niewłaściwy typ.

2.2.1.3 Historia źródła danych

Aby uruchomić funkcję „Historia źródła danych”, należy w zakładce „Zarządzanie źródłami danych”, wybrać:

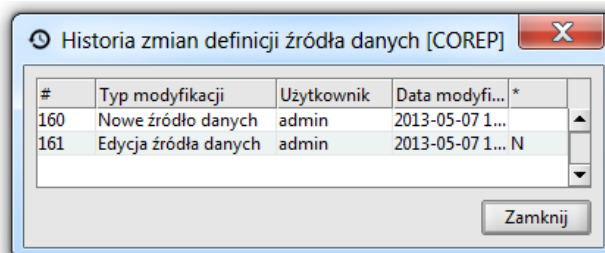
- **Menu „Źródło danych”** →  Historia źródła danych

lub

- Ikonę:  .

Funkcja ta:

- wyświetla informację o zmianach wprowadzonych do wybranego (podświetlonego) źródła danych,
- prezentuje historię wprowadzonych zmian (zob. Rysunek 64):



#	Typ modyfikacji	Uzytkownik	Data modyfi...	*
160	Nowe źródło danych	admin	2013-05-07 1...	
161	Edycja źródła danych	admin	2013-05-07 1... N	

Rysunek 64. Historia zmian definicji źródła danych

2.2.1.4 Importuj źródła danych

Aby uruchomić funkcję „Importu definicji źródła danych”, należy w zakładce „Zarządzanie źródłami danych”, wybrać:

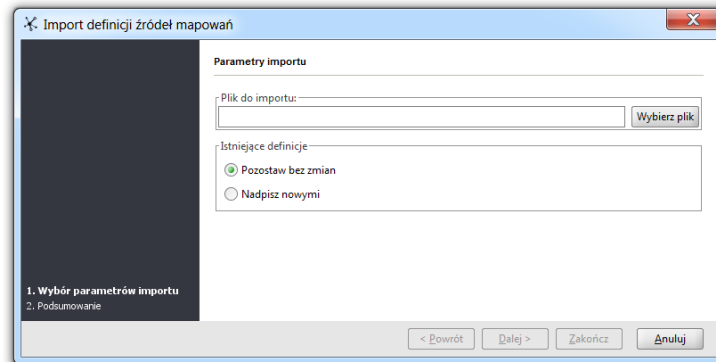
- **Menu „Źródło danych”** →  Importuj źródła danych

lub

- Ikonę:  .

Funkcja ta:

- pozwala na wykonanie importu definicji źródeł danych (zob. Rysunek 65):



Rysunek 65. Import definicji źródeł mapowań

Plik z importowanym źródłem danych typu słownikowego zawiera:

- podstawowe źródło danych
- źródła zdefiniowane w słownikach.

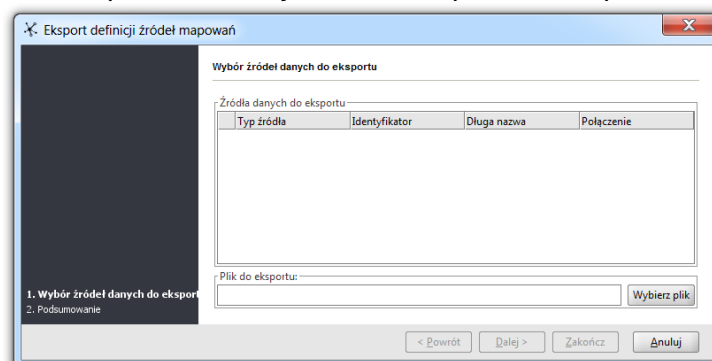
2.2.1.5 Eksportuj źródła danych

Aby uruchomić funkcję „Eksportuj źródła danych”, należy w zakładce „Zarządzanie źródłami danych”, wybrać:

- Menu „Źródło danych” →  Eksportuj źródła danych
- lub
- Ikonę: .

Funkcja ta:

- pozwala na wykonanie eksportu definicji źródeł danych (zob. Rysunek 66):





Rysunek 66. Eksport definicji źródeł mapowań

Plik z eksportowanym źródłem danych typu słownikowego zawiera:

- podstawowe źródło danych
- oraz:
- źródła zdefiniowane w słownikach.
 -

2.2.1.6 Import zbioru mapowań wraz z używanymi definicjami źródeł

Aby uruchomić funkcję „Import zbioru mapowań wraz z używanymi definicjami źródeł do paczki”, należy:


- na bocznej zakładce „Zarządzanie zbiorami” przejść do widoku listy (skrót klawiszowy Ctrl-I). Następnie z menu „Zbiór mapowań” wybrać opcję  **Importuj paczkę mapowań** lub użyć ikony .

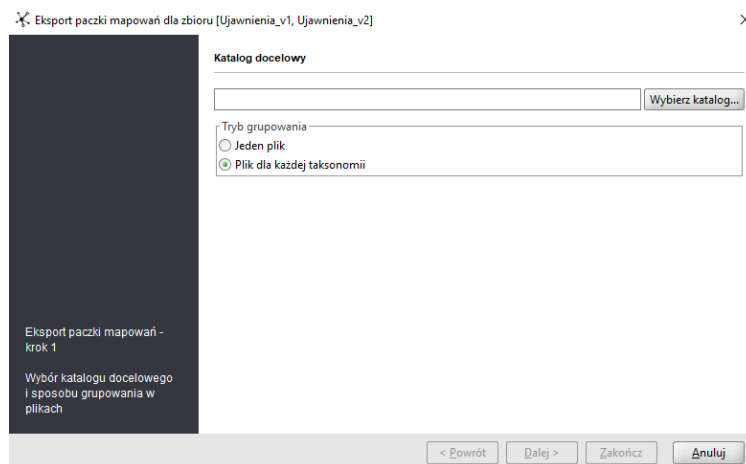
Na wyświetlonym ekranie należy wybrać plik.

Funkcja ta pozwala na zaimportowanie zbioru mapowań wraz z używanymi definicjami źródeł.

2.2.1.7 Eksport zbioru mapowań wraz z używanymi definicjami źródeł

Aby uruchomić funkcję „Eksport zbioru mapowań wraz z używanymi definicjami źródeł do paczki”, należy:

- na bocznej zakładce „Zarządzanie zbiorami” przejść do widoku listy (skrót klawiszowy Ctrl-I). Następnie zaznaczyć wybrany zbiór (lub zbiory) mapowań i użyć ikony .



Rysunek 67. Eksport zbioru mapowań wraz z używanymi definicjami źródeł

Na wyświetlonym ekranie należy wybrać:

- katalog zapisu pliku,
- tryb grupowania (opcja dostępna tylko przy wyborze więcej niż jednego zbioru mapowań).

Funkcja ta pozwala na wyeksportowanie zbioru mapowań wraz z używanymi definicjami źródeł.

Plik z wyeksportowanym zbiorem mapowań wraz z używanymi definicjami zawiera:

- definicje zbiorów mapowań,
- źródła danych.

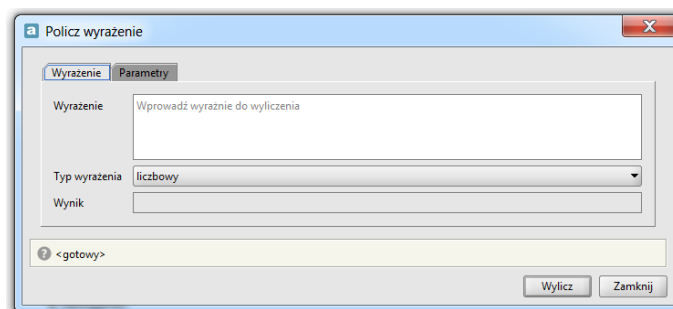
2.2.1.8 Policz wyrażenie

Aby uruchomić funkcję „Policz wyrażenie” należy, w zakładce „Zarządzanie źródłami danych”, wybrać:

- **Menu „Źródło danych”** → **Policz wyrażenie** .

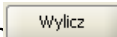
Funkcja ta:

- pozwala na sprawdzenie poprawności zdefiniowanych wyrażeń mapujących i ich źródeł zasilania;
- szybkie wyliczenie danych w oparciu o sformułowane wyrażenie i źródło danych (zob. Rysunek 68):



Rysunek 68. Szybkie sprawdzenie definiowanych wyrażeń do zasilania danymi

Po

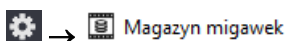
- wprowadzeniu do wiersza:
 - wyrażenia do zasilania danymi,
- wybraniu odpowiedniego typu wyrażenia:
 - liczbowy,
 - logiczny,
 - napis,
- zaakceptowaniu wprowadzonych informacji klawiszem  ,

w oparciu o podane w tym wyrażeniu źródło danych, **system dokona sprawdzenia i przeliczenia danych**.



W przypadku pojawienia się błędów w wyliczaniu, użytkownik zostanie o tym poinformowany stosownym komunikatem.

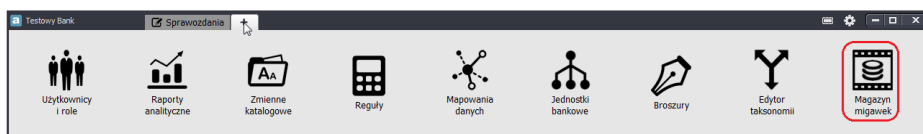
3 Moduł „Magazyn migawek”

Aby uruchomić „Magazyn migawek”, należy wybrać:



lub

 - znajdujący się na pasku zakładek modułowych (zob. Rysunek 69) i z wyświetlonej listy wybrać:  .



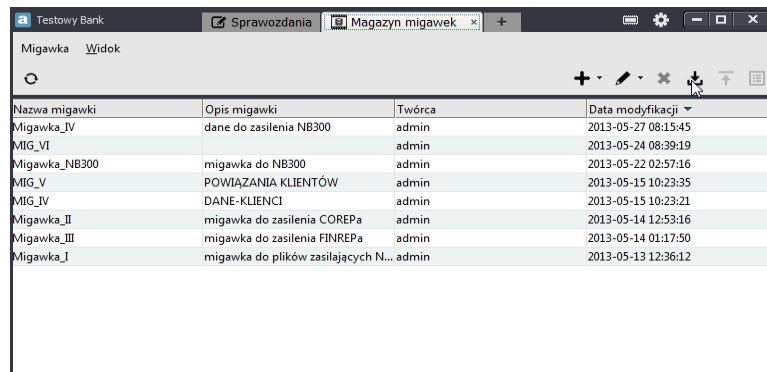
Rysunek 69. Lista zakładek modułowych aSIS – z wyróżnionym modułem „Magazyn migawek”

Utworzy się wówczas nowa zakładka modułowa  :

- ściśle powiązana z modułem mapowań,
- dzięki której możliwe jest znaczne rozszerzenie możliwości modułu mapowań o migawkowe źródła danych,
- pozwalająca na :
 - tworzenie i zarządzanie migawkami, będącymi zbiorami tabel zbudowanych na podstawie danych źródłowych i przekształceń SQL, przechowywanych w niezależnej bazie danych wbudowanej w aplikację aSIS;
 - przechowywanie danych źródłowych osobno, dla każdego okresu sprawozdawczego w formie instancji migawek.

Po uaktywnieniu „Magazynu migawek” (zob. Rysunek 70):

- wyświetlana jest lista wszystkich wcześniej zdefiniowanych definicji migawek,
- dostępne są funkcje:
 - pogrupowane w menu „Migawka” i „Widok”,
 - prezentowane w formie ikon funkcyjnych.



Nazwa migawki	Opis migawki	Twórca	Data modyfikacji
Migawka_IV	dane do zasilenia NB300	admin	2013-05-27 08:15:45
MIG_VI		admin	2013-05-24 08:39:19
Migawka_NB300	migawka do NB300	admin	2013-05-22 02:57:16
MIG_V	POWIĄZANIA KLIENTÓW	admin	2013-05-15 10:23:35
MIG_IV	DANE-KLIENCI	admin	2013-05-15 10:23:21
Migawka_II	migawka do zasilenia COREPa	admin	2013-05-14 12:53:16
Migawka_III	migawka do zasilenia FINREPa	admin	2013-05-14 01:17:50
Migawka_I	migawka do plików zasilających N...	admin	2013-05-13 12:36:12

Rysunek 70. Magazyn migawek – prezentacja listy migawek

Moduł “Magazyn migawek”:

- dostępny jest wyłącznie dla banków, posiadających licencję rozszerzoną o ten moduł;
- przy wersji wielostanowiskowej może być obsługiwany jednocześnie wyłącznie na jednym stanowisku.

Migawki:

- to definicje tabelarycznych danych źródłowych, wykorzystywane w module mapowań jako migawkowe źródła danych;
- mogą się składać z wielu różnych, niezależnych tabel, które zasilane są danymi z systemów backoffice ;
- nie są powiązane z żadnymi okresami sprawozdawczymi,
- nie definiują żadnych powiązań pomiędzy różnymi tabelami źródłowymi,
- identyfikowanymi są poprzez ich unikatowe nazwy.

Prezentowana lista definicji migawek:

- dostarcza informacji o:
 - nazwie migawki,
 - zastosowanym opisie,
 - loginie twórcy,
 - dacie modyfikacji,
- dla większej przejrzystości, może zostać posortowana według dowolnej kolumny.

Więcej informacji na temat zdefiniowanych migawek można uzyskać przeglądając je przy wykorzystaniu funkcji „Edytuj definicje migawek” (zob. rozdział [Edytuj definicje migawki](#)).

W magazynie migawek w ten sam sposób prezentowane są wszystkie wcześniej zdefiniowane migawki, niezależnie od metody przyjętej przy ich definiowaniu, czyli:

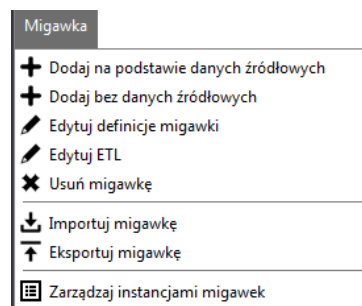
- „Dodaj na podstawie danych źródłowych” (zob. rozdział [Dodaj na podstawie danych źródłowych](#)),

- „Dodaj bez danych źródłowych” (zob. rozdział [Dodaj bez danych źródłowych](#)).

3.1 Migawka

Menu „Migawka” – grupuje funkcje (zob. Rysunek 71) pozwalające na manipulowanie definicjami migawek, czyli:

- tworzenie nowych migawek,
- edycję definicji migawek wcześniej utworzonych,
- edycję przekształceń ETL,
- usuwanie niepotrzebnych i błędnie zdefiniowanych migawek;
- import / eksport migawek,
- zarządzanie instancjami migawek.




Rysunek 71. Menu „Migawka” w module „Magazyn migawek”

Z wyjątkiem funkcji importu i tworzenia nowych migawek, wszystkie pozostałe funkcje uaktywniają się dopiero po podświetleniu (zaznaczeniu) na liście migawki, dla której chcemy uruchomić wymaganą funkcję.


3.1.1 Dodaj na podstawie danych źródłowych

Aby utworzyć „Nową migawkę na podstawie danych źródłowych”, należy wybrać:

- **Menu „Migawka”** →  Dodaj na podstawie danych źródłowych

lub

- Ikonę: ,

a następnie opcję  Dodaj na podstawie danych źródłowych

bądź

- skrót klawiszowy Ctrl+Shift+A.

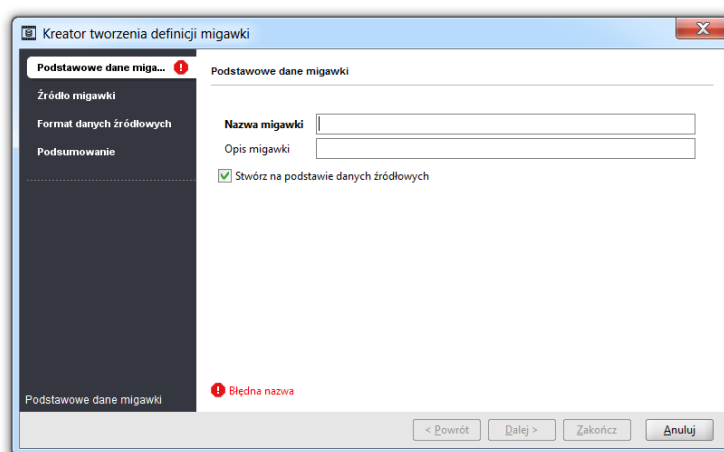
Funkcja ta:

- w oparciu o tabelaryczne dane źródłowe zapisane w formacie CSV, pozwala na szybkie definiowanie migawek.

Kreator tworzenia migawki każdorazowo przebiega w 4 krokach.

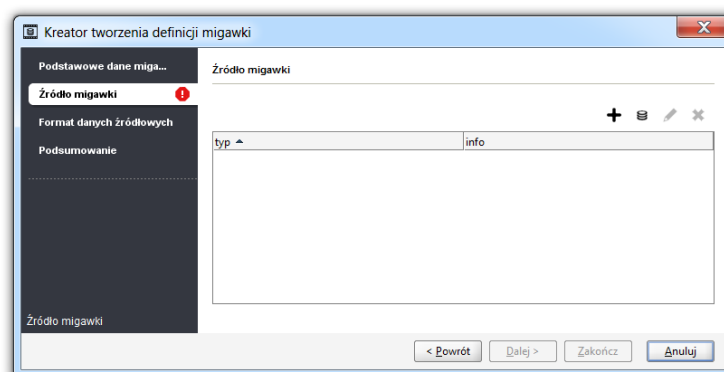
Krok 1 – to określenie danych podstawowych tworzonej migawki, czyli podanie (zob. Rysunek 72):

- nazwy migawki (max 125 znaków),
- opisu migawki (max 255 znaków) – opcjonalnie.




Rysunek 72. Wybór danych podstawowych przy tworzeniu migawki

Krok 2 – to określenie źródeł dla tworzonej migawki (zob. Rysunek 73).




Rysunek 73. Określenie źródeł dla nowotworzonej migawki


Poprzez dostępne tu klawisze funkcyjne, możliwe jest:

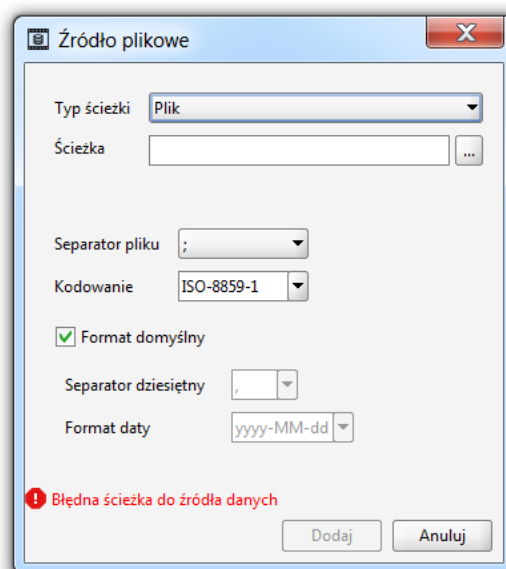
- + - wskazanie:
 - pliku,
 - katalogu,
-  - wskazanie tabel z bazy danych,

 - edytowanie wcześniej wskazanego źródła,

 - usunięcie wcześniej wskazanego źródła.

Po wybraniu funkcji „Dodaj źródło plikowe”, poprzez przycisk funkcji , wyświetla się ekran (zob. Rysunek 74) na którym:

- po dokonaniu wyboru typu ścieżki źródła danych:
 - plik,
 - katalog,
- wymagane jest przy użyciu przycisku , wskazanie ścieżki dostępu do źródła.




Rysunek 74. Definiowanie źródła plikowego

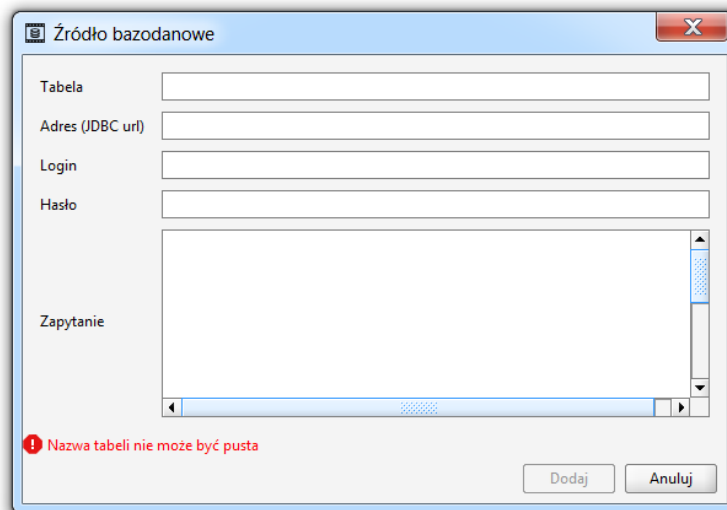
Dodatkowo, można wprowadzić zmiany w prezentowanych ustawieniach:

- separatora pliku (; , I <TAB>),
- sposobu kodowania:
 - ISO 8859-1,
 - ISO-8859-2,
 - Windows-1250,
 - UTF-8,
 - UTF-16,
 - US-ASCII,
- separatora dziesiętnego (lub .),
- formatu daty:
 - yyyy-MM-dd,
 - yyyy.MM.dd,
 - dd-MM-yyyy,

- o dd.MM.yyy,

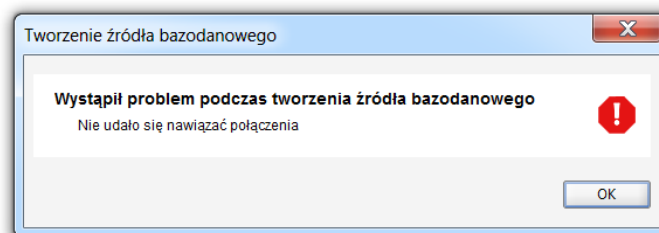
Po wybraniu funkcji „Dodaj źródło bazodanowe”, poprzez przycisk funkcji , wyświetla się ekran (zob. Rysunek 75), na którym wymagane jest podanie:

- nazwy tabeli,
- adresu (JDBC url),
- danych dostępowych do bazy (login, hasło),
- zapytania SQL.



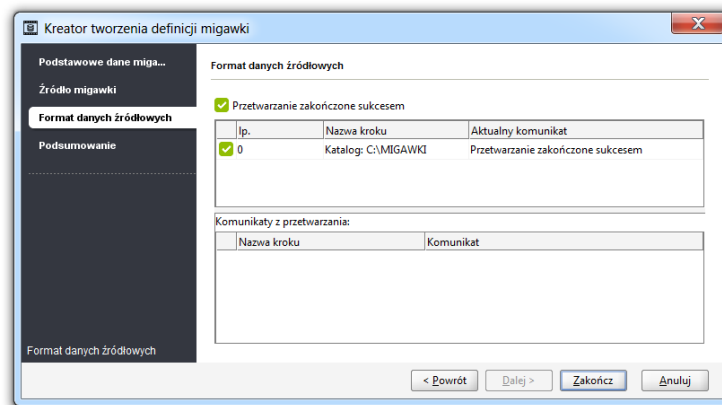
Rysunek 75. Dodanie źródła bazodanowego

W przypadku podania niewłaściwych danych, może zostać wyświetlony komunikat (zob. Rysunek 76):



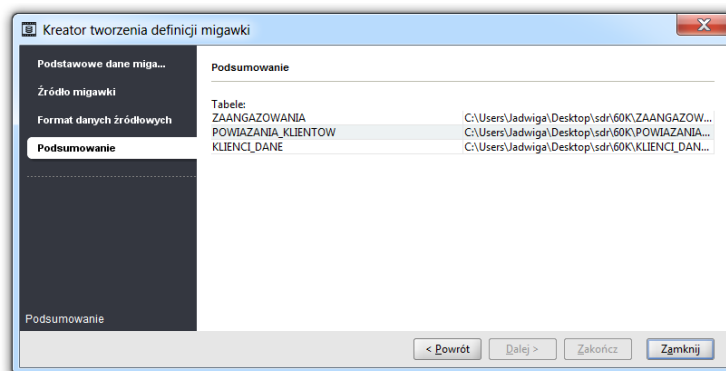
Rysunek 76. Komunikat o błędnych danych tworzonego źródła bazodanowego

Krok 3 – to wstępna walidacja dokonanych wyborów (zob. Rysunek 77).



Rysunek 77. Wstępna walidacja wybranych danych źródłowych

Krok 4 – to prezentacja tabel z którymi została połączona utworzona migawka (zob. Rysunek 78).




Rysunek 78. Prezentacja tabel, z którymi została połączona utworzona migawka

Wybranie przycisku  spowoduje:

- zakończenie działania funkcji,
- zapisanie zdefiniowanej migawki, zgodnie z wyborami dokonanymi podczas jej tworzenia;
- dopisanie nowoutworzonej migawki do listy zdefiniowanych migawek (zob. Rysunek 70).


3.1.2 Dodaj bez danych źródłowych

Aby utworzyć „Nową migawkę bez wzorowania się na danych źródłowych”, należy wybrać:

- Menu „Migawka” →  Dodaj bez danych źródłowych

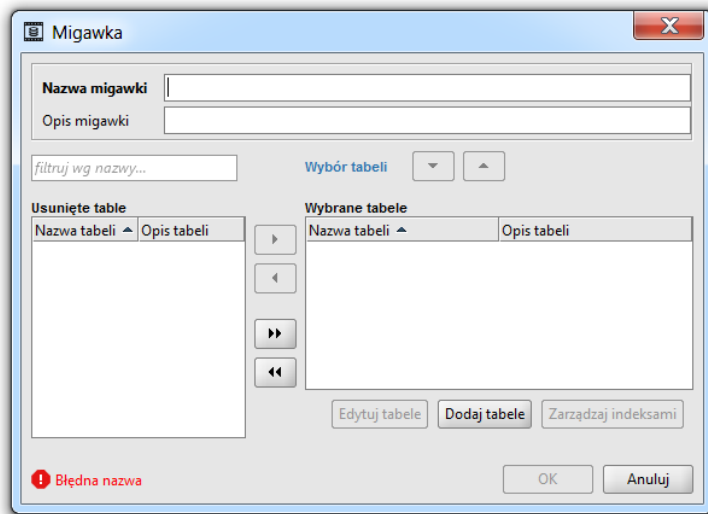
lub

- Ikonę: ,

a następnie opcję  Dodaj bez danych źródłowych .

Funkcja ta:

- wymaga ręcznego zdefiniowania tabel, z którymi ma być powiązana tworzona migawka;
- jest wykorzystywana, gdy tworzona migawka odwołuje się do wielu różnych danych źródłowych i przez to definiowanie migawki na podstawie jednego źródła jest mało wygodne.



Rysunek 79. Tworzenie nowej migawki, bez wskazywania danych źródłowych

Obok danych podstawowych, takich jak (Rysunek 79:):

- nazwa migawki (max 125 znaków),
- opis migawki (max 255 znaków) – opcjonalnie,

konieczne jest tutaj:

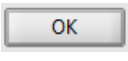
- zdefiniowanie tabel powiązanych z tworzoną migawką (zob. rozdział [Dodaj nową tabelę](#)),

możliwe jest:

- wprowadzanie zmian do błędnie zdefiniowanych tabel (zob. rozdział [Edytuj tabelę](#)),
- definiowanie i zarządzanie indeksami, usprawniającymi szybsze wykonywanie zapytań SQL (zob. rozdział [Zarządzaj indeksami](#)).


Lista wszystkich zdefiniowanych tabel, prezentowana jest zawsze w prawym panelu okna.

Na liście tej powinny się znajdować wszystkie te tabele, które są powiązane z tworzoną migawką.

Po zdefiniowaniu tabeli uaktywnia się przycisk  i możliwe jest zapisanie utworzonej migawki i jej prezentację na liście (zob. Rysunek 70).


3.1.2.1 Dodaj nową tabelę

Aby zdefiniować migawkę przy wykorzystaniu funkcji „Dodaj bez danych źródłowych”, konieczne jest zdefiniowanie wszystkich tabel powiązanych z tą migawką, przy wykorzystaniu funkcji „Dodaj tabelę” (zob. Rysunek 79), czyli wybierając:

- Menu „Migawka” →  Dodaj bez danych źródłowych

lub

- Ikonę: 

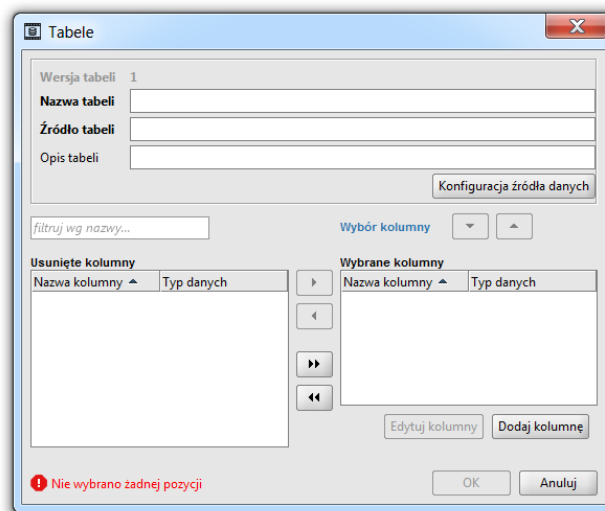
i opcję:  Dodaj bez danych źródłowych ,

a następnie

- przycisk funkcyjny:  .

Funkcja ta:

- pozwala na zdefiniowanie nowej tabeli,
- wyświetla ekran (zob. Rysunek 80).



Rysunek 80. Ręczne definiowanie tabel do nowotworzonej migawki

Wymagane jest tu podanie:

- nazwy tabeli (max 125 znaków),
- źródła tabeli,
- opisu tabeli (max 255 znaków) – opcjonalnie,
- szczegółowych informacji o kolumnach występujących w tabeli (zob. Rysunek 77).

Dla wskazywanego źródła danych, konieczne jest dodatkowo podanie danych konfiguracyjnych tego tabelarycznego źródła danych.

3.1.2.1.1 Konfiguracja danych źródłowych

Aby ustawić odpowiednią konfigurację źródła danych, należy wykorzystać funkcję „Konfiguracja danych źródłowych” poprzez wybranie:

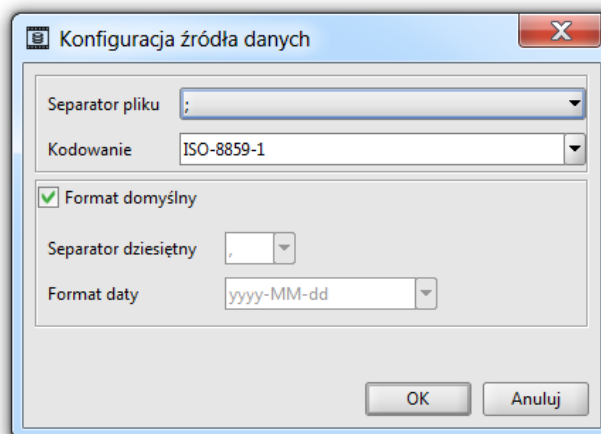
- przycisku funkcyjnego Konfiguracja źródła danych.

Wyświetli się wówczas ekran (zob. Rysunek 81), na którym możliwe jest wprowadzenie zmian w zakresie:

- stosowanych separatorów pliku:
 - ;
 - ,
 - |
 - <TAB>
- sposobu kodowania znaków:
 - ISO 8859-1,
 - ISO-8859-2,
 - Windows-1250,
 - UTF-8,
 - UTF-16,
 - US-ASCII,
- wykorzystywanego separatora dziesiętnego:
 - , (format domyślny),
 - .
- formatu dat:
 - yyyy-MM-dd (format domyślny),
 - yyyy-MM.dd,
 - dd-MM-yyy,
 - dd.MM.yyy.

Zmiany w zakresie wykorzystywanego separatora dziesiętnego i formatu dat mogą zostać

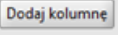
wprowadzone po wcześniejszym wyłączeniu opcji Format domyślny, czyli przy opcji Format domyślny.

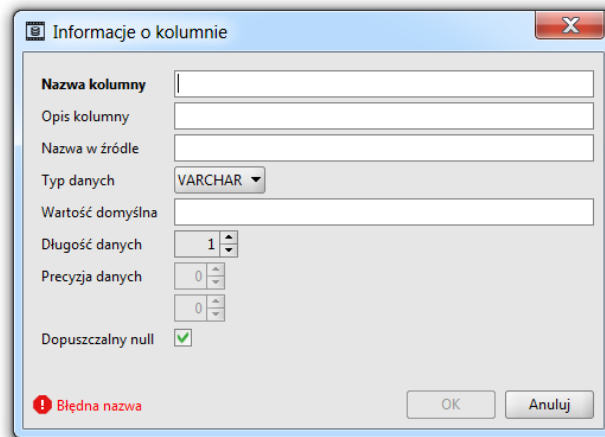


Rysunek 81. Konfiguracja źródła danych dla definiowanej tabeli

3.1.2.1.2 Dodaj kolumnę

Aby zdefiniować kolumny, występujące w tabelarycznym źródle danych, powiązanych z tworzoną migawką, należy uruchomić funkcję „Dodaj kolumnę”, wybierając:

- przycisk funkcyjny:  (zob. Rysunek 82).



Rysunek 82. Informacje o kolumnie definiowanej tabeli

Funkcja ta wymaga podania:


- nazwy kolumny (max 125 znaków),
- nazwy w źródle

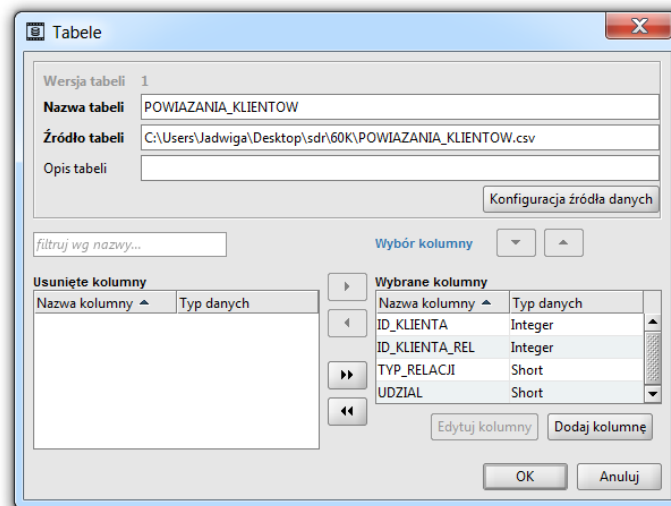
i w zależności od wybranego typu danych, wybranie odpowiedniej:

- długości danych,
- precyzji danych.

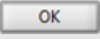
Typ danych, może zostać wybrany jako:

- VARCHAR – znakowy o określonej długości,
- INTEGER – całkowito-liczbowy średni,
- SHORT – całkowito-liczbowy krótki,
- LONG – całkowito-liczbowy długi,
- DECIMAL – stało-przecinkowy,
- DATE – typu data.

Zatwierdzenie wprowadzonych ustawień przyciskiem , spowoduje zapisanie zdefiniowanej kolumny w **prawym panelu prezentowanego** okna (zob. Rysunek 83).

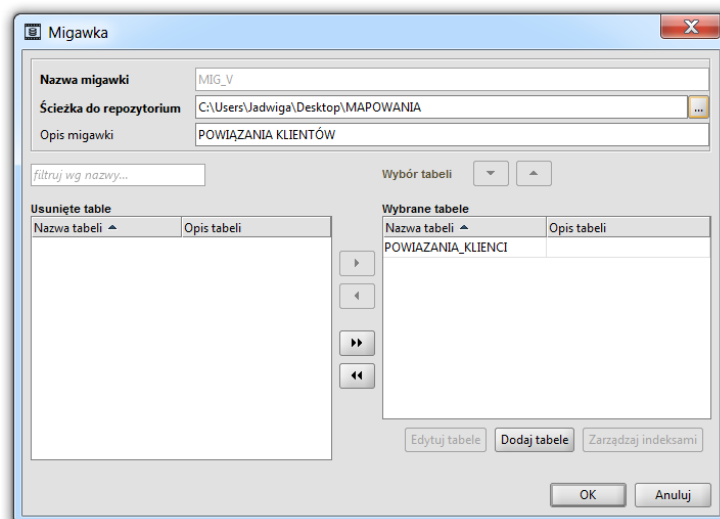


Rysunek 83. Lista wybranych kolumn dla definiowanej tabeli

Wymagane jest tu zdefiniowanie wszystkich kolumn dla definiowanej tabeli tworzonej migawki i zatwierdzenie dokonanych wyborów przyciskiem  .

Aplikacja:

- powróci wówczas do ekranu, na którym prezentowana jest lista tabel powiązanych z tworzoną migawką (por. Rysunek 79 i Rysunek 84);
- umożliwi dodanie kolejnej tabeli do tworzonej migawki.



Rysunek 84. Definiowanie migawki – zdefiniowanie i zapisanie wymaganej tabeli

3.1.2.1.3 Usuń kolumnę

Jeśli z jakiegoś powodu, na liście zdefiniowanych kolumn znalazły się takie kolumny, które nie powinny być powiązane z daną tabelą, to możliwe jest ich usunięcie.

Aby usunąć wcześniej zdefiniowane kolumny, należy:

- przenieść usuwane kolumny z prawego do lewego panelu okna.

Można to wykonać przy wykorzystaniu przycisków funkcyjnych:



- usunięcie jednej lub kilku zaznaczonych kolumn tabeli,



- usunięcie wszystkich zdefiniowanych kolumn,

Usunięcie zdefiniowanych kolumn i przeniesienie ich do lewego panelu okna nie musi być czynnością trwałą, gdyż w dowolnym momencie możliwe jest ich ponowne włączenie, przy wykorzystaniu przycisków:



- włączenie wszystkich kolumn wcześniej przeniesionych do lewego panelu okna,



- włączenie wybranych (zaznaczonych) kolumn wcześniej przeniesionych do lewego panelu okna.

3.1.2.1.4 Edytuj kolumnę

Zdefiniowane kolumny, znajdujące się na liście prawego panelu okna (zob. Rysunek 79), mogą być modyfikowane przy wykorzystaniu funkcji: „Edytuj kolumny” poprzez użycie :

- przycisku funkcyjnego:  .

Funkcja ta:

- uaktywnia się po podświetleniu (zaznaczeniu) kolumny, dla której mają zostać wprowadzone zmiany;
- wyświetla dane wcześniej wprowadzone,
- pozwala na wprowadzenie zmian we wszystkich prezentowanych pozycji.


3.1.2.2 Usuń tabelę

Jeśli z jakiegoś powodu na liście wybranych tabel dla tworzonej migawki znalazły się tabele, które nie powinny być powiązane z tworzoną migawką, to możliwe jest ich usunięcie (według tej samej zasady jak usuwanie błędnych kolumn).

Aby usunąć wcześniej zdefiniowane tabele, należy:

- przenieść usuwane tabele z prawego do lewego panelu okna.

Zastosowane tu przyciski funkcyjne mają za zadanie:

 - usunięcie jednej lub kilku zaznaczonych tabel,

 - usunięcie wszystkich zdefiniowanych tabel.


Wyłączenie tabel z migawki i przeniesienie ich do lewego panelu okna nie musi być czynnością trwałą, gdyż w dowolnym momencie możliwe jest ich ponowne włączenie, przy wykorzystaniu przycisków:

 - włączenie wszystkich tabel wcześniej przeniesionych do lewego panelu okna,

 - włączenie wybranych (zaznaczonych) tabel wcześniej przeniesionych do lewego panelu okna.


3.1.2.3 Edytuj tabelę

Podczas definiowania migawki przy wykorzystaniu funkcji „Dodaj bez danych źródłowych”, możliwe jest wprowadzanie zmian do zdefiniowanych a jeszcze nie zapisanych tabel, przy wykorzystaniu funkcji „Edytuj tabelę” (zob. Rysunek 84), czyli wybierając:

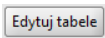
- Menu „Migawka” →  Dodaj bez danych źródłowych

lub

- Ikonę: 

i opcję:  Dodaj bez danych źródłowych ,

a następnie po zdefiniowaniu tabeli:

- podświetlenie w prawym panelu okna tabeli, dla której mają zostać wprowadzone jakieś zmiany i wybranie
- przycisku funkcyjnego:  .

Funkcja ta:




- uaktywnia się dopiero po podświetleniu (zaznaczeniu) w prawym panelu okna tabeli, dla której mają zostać wprowadzone zmiany;
- prezentuje ekran z wcześniej wprowadzonymi danymi (zob. Rysunek 83),

- pozwala na modyfikowanie wszystkich pozycji.

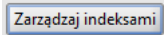
3.1.2.4 Zarządzaj indeksami

W celu szybszego wykonywania zdefiniowanych zapytań SQL, możliwe jest wykorzystanie funkcji „Zarządzaj indeksami” i przydzielenie wybranym kolumnom w tabelarycznych danych źródłowych dodatkowych indeksów.

Funkcja ta może zostać uaktywniona przy definiowaniu migawki poprzez funkcję „Dodaj bez danych źródłowych” (zob. Rysunek 80), przez wybranie:

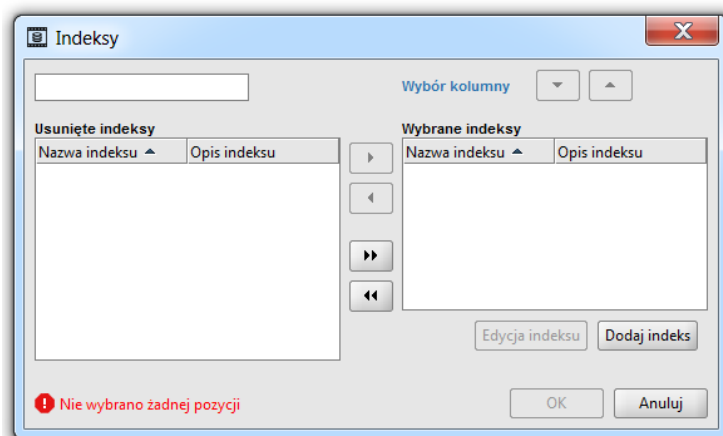
- Menu „Migawka” →  Dodaj bez danych źródłowych
- lub
- Ikonę: ,
- i opcję:  Dodaj bez danych źródłowych ,

a następnie po zdefiniowaniu tabeli:

- podświetlenie w prawym panelu okna tabeli, dla której ma zostać uaktywnione „Zarządzanie indeksami”
- i wybranie
- przycisku funkcyjnego:  .

Funkcja ta:

- uaktywnia się dopiero po podświetleniu (zaznaczeniu) w prawym panelu okna tabeli, dla której ma zostać uaktywnione „Zarządzanie indeksami”;
- pozwala na definiowanie i usuwanie błędnych indeksów (zob. Rysunek 85):

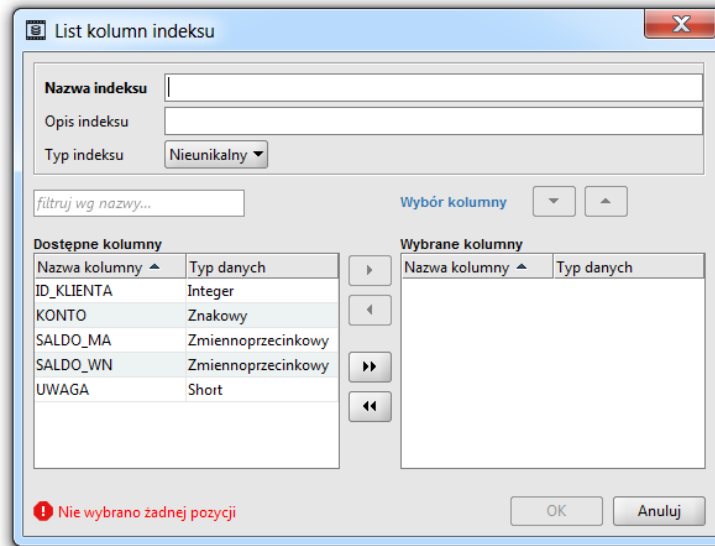


Rysunek 85. Zarządzanie indeksami

3.1.2.4.1 Dodaj indeks

Aby zdefiniować indeksy dla kolumn wybranej tabeli, należy uruchomić funkcję „Dodaj indeks”, wybierając:

- przycisk funkcyjny:  (zob. Rysunek 86):



Rysunek 86. Lista kolumn indeksu

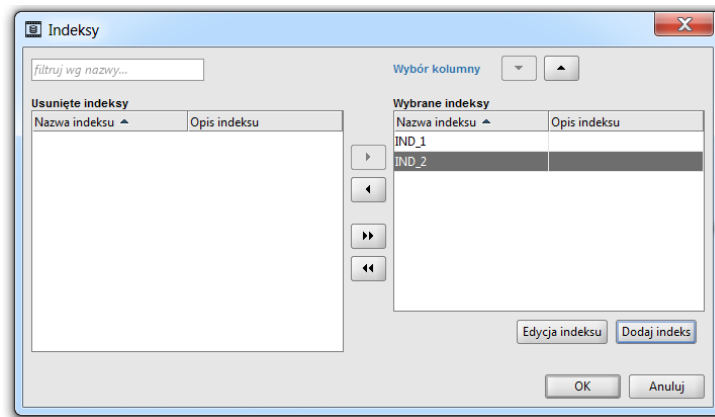
Na ekranie w lewym panelu okna prezentowane są wszystkie wcześniej zdefiniowane, dostępne kolumny dla wybranej tabeli.

Aby zdefiniować indeksy, wymagane jest tutaj:

- podanie nazwy tworzonego indeksu (max znaków),
- wybranie odpowiedniego typu indeksu:
 - unikalny,
 - nieunikalny,
- przeniesienie z lewego do prawego panelu okna tych kolumn, dla których ma zostać przydzielony tworzony indeks.

Zatwierdzenie dokonanych wyborów przyciskiem  spowoduje:

- nadanie wybranym kolumnom dodatkowych indeksów,
- zapisanie zdefiniowanych indeksów dla danej migawki,
- prezentację utworzonych indeksów w prawym panelu wyświetlanego okna (zob. Rysunek 87):




Rysunek 87. Zarządzanie indeksami - lista utworzonych indeksów


3.1.2.4.2 Usuń indeks

Aby usunąć wcześniej zdefiniowany indeks, należy:

- przenieść wybrany indeks z prawego do lewego panelu okna.


Można to wykonać przy wykorzystaniu przycisków funkcyjnych:

 - usunięcie jednego lub kilku zaznaczonych indeksów,

 - usunięcie wszystkich zdefiniowanych indeksów.

Usunięcie zdefiniowanych indeksów i przeniesienie ich do lewego panelu okna nie musi być czynnością trwałą, gdyż w dowolnym momencie możliwe jest ich ponowne włączenie, przy wykorzystaniu przycisków:

 - włączenie wszystkich indeksów wcześniej przeniesionych do lewego panelu okna,

 - włączenie wybranych (zaznaczonych) indeksów wcześniej przeniesionych do lewego panelu okna.

3.1.2.4.3 Edytuj indeks

Zdefiniowane indeksy, znajdujące się na liście prawego panelu okna (zob. Rysunek 87), mogą być modyfikowane przy wykorzystaniu funkcji: „Edycja indeksu” poprzez użycie :




- przycisku funkcyjnego:  .

Funkcja ta:

- uaktywnia się po podświetleniu (zaznaczeniu) indeksu, dla którego mają zostać wprowadzone zmiany;
- wyświetla dane wcześniej wprowadzone,
- pozwala na wprowadzenie zmian we wszystkich prezentowanych pozycji.

3.1.3 Edytuj definicje migawki

Aby wprowadzić zmiany do wcześniej zdefiniowanej migawki, należy wykorzystać funkcję „Edytuj definicje migawki”, wybierając:

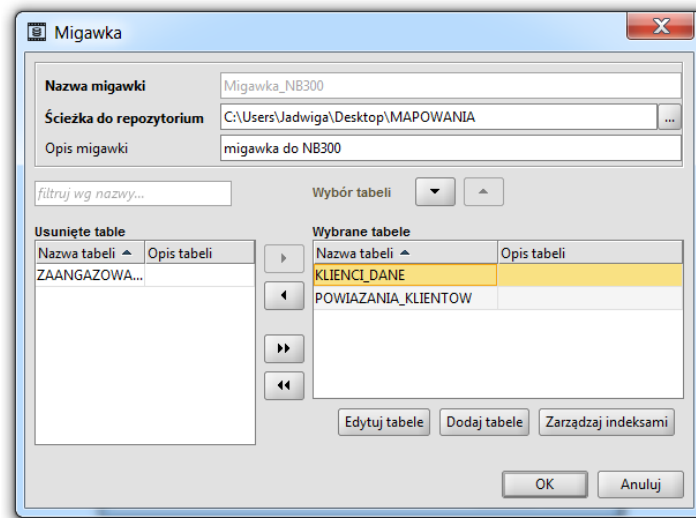
- **Menu „Migawka”** →  Edytuj definicje migawki
- lub
- ikonę: 
- i opcję:  Edytuj definicje migawki
- lub
- skrót klawiszowy Ctrl-W.

Funkcja ta:

- pozwala na wprowadzenie zmian w definicjach utworzonych migawek,
- uaktywnia się, po zaznaczeniu (podświetleniu) na liście migawki, dla której mają zostać wprowadzone modyfikacje;
- dostarcza szczegółowych informacji o zdefiniowanych i zapisanych migawkach;
- dotyczy wszystkich zdefiniowanych migawek, niezależnie od tego w jaki sposób były one utworzone:
na podstawie danych źródłowych, czy bez danych źródłowych.

Po jej uruchomieniu:

- prezentowany jest ekran zawierający wcześniej wprowadzone dane (zob Rysunek 88 por. Rysunek 84),
- możliwe jest:
 - dodanie nowych tabel,
 - usunięcie błędnych tabel,
 - ponowne dodanie tabel wcześniej usuniętych (znajdujących się w lewym panelu okna),
 - wprowadzenie zmian do wybranych, prezentowanych w prawym panelu okna tabel;
- może zostać zmieniony opis migawki,
- nie można zmienić nazwy migawki.



Rysunek 88. Edycja definicji migawki


3.1.3.1 Dodaj nową tabelę

Aby wcześniej zdefiniowaną migawkę rozszerzyć o nowe tabelaryczne źródło danych, należy wykorzystać funkcję „Dodaj table”, wybierając:

- Menu „Migawka” →  Edytuj definicje migawki

lub

- Ikone: 

i opcję:  Edytuj definicje migawki ,

a następnie

- przycisk funkcyjny:  .


Funkcja ta:

- pozwala na zdefiniowanie nowej tabeli,
- po uruchomieniu prezentuje ekran (zob. Rysunek 80),
- ma taki sam przebieg i działanie, jak uruchomiona przy definiowaniu migawki poprzez funkcję „Dodaj bez danych źródłowych” (zob. rozdział [Dodaj nową tabelę](#)).

3.1.3.2 Usuń / przywróć tabelę

Aby usunąć wcześniej zdefiniowane table, należy:

- przenieść usuwane table z prawego do lewego panelu okna,
- wykorzystać do tego jeden z przycisków funkcyjnych:

 - usunięcie jednej lub kilku zaznaczonych tabel,

 - usunięcie wszystkich zdefiniowanych tabel.

Aby przywrócić wcześniej zdefiniowane a później usunięte tabele, należy:

- przenieść przywracane tabele z lewego do prawego panelu okna,
- wykorzystać do tego jeden z przycisków funkcyjnych:

 - włączenie wszystkich tabel wcześniej przeniesionych do lewego panelu okna,

 - włączenie wybranych (zaznaczonych) tabel wcześniej przeniesionych do lewego panelu okna.

Funkcje te:

- mają takie samo działanie, jak uruchomione przy definiowaniu migawki poprzez funkcję „Dodaj bez danych źródłowych” (zob. rozdział [Usuń tabelę](#)).

3.1.3.3 Edytuj tabelę


Aby wprowadzić zmiany do tabel wcześniej zdefiniowanej migawki, należy wykorzystać funkcję „Edytuj tabelę”, wybierając:

- **Menu „Migawka”** →  Edytuj definicje migawki

lub

- Ikonę: ,
- i opcję:  Edytuj definicje migawki

a następnie

- podświetlenie w prawym panelu okna tabeli, dla której mają zostać wprowadzone jakieś zmiany i wybranie
- przycisku funkcyjnego 

Funkcja ta:

- pozwala na modyfikację wszystkich ustawień dla wcześniej zdefiniowanych tabel,
- po uruchomieniu prezentuje ekran z wcześniej wprowadzonymi danymi (zob. Rysunek 83),
- ma taki sam przebieg i działanie, jak uruchomiona przy definiowaniu migawki poprzez funkcję „Dodaj bez danych źródłowych” (zob. rozdział [Edytuj tabelę](#)).


3.1.3.4 Zarządzaj indeksami

Aby przyspieszyć wykonywanie zdefiniowanych zapytań SQL, można wybranym kolumnom w tabelarycznych danych źródłowych przy wykorzystaniu funkcji „Zarządzaj indeksami” nadać dodatkowe indeksy, wybierając:

- **Menu „Migawka”** →  Edytuj definicje migawki

lub

- ikonę: 

i opcję:  Edytuj definicje migawki ,

a następnie

- podświetlenie w prawym panelu okna tabeli, dla której ma zostać uaktywnione „Zarządzanie indeksami”

i wybranie

- przycisku funkcyjnego:  .

Funkcja ta:

- pozwala na definiowanie i usuwanie błędnych indeksów;
- po uruchomieniu prezentuje ekran (zob. Rysunek 87),
- ma taki sam przebieg i działanie, jak uruchomiona przy definiowaniu migawki poprzez funkcję „Dodaj bez danych źródłowych” (zob. rozdział [Dodaj bez danych źródłowych](#)).

3.1.4 Edytuj ETL

Aby uruchomić funkcję „Edytuj ETL”, należy wybrać:

- **Menu „Migawka”** →  Edytuj ETL

lub

- ikonę: 

i opcję:  Edytuj ETL

lub

- skrót klawiszowy Ctrl+Shift+W.

Funkcja ta:

- pozwala na wykonanie szeregu przekształceń w utworzonych migawkach,
- uaktywnia się, po zaznaczeniu (podświetleniu) na liście migawki, dla której mają zostać wprowadzone modyfikacje.

Po jej uruchomieniu:

- prezentowany jest ekran zawierający tabele migawkowe zdefiniowane w wybranej migawce (zob. Rysunek 84).

ETL jest narzędziem wykorzystującym wiele złożonych funkcji i wymaga szerszego omówienia (zob. rozdział [ETL](#)).

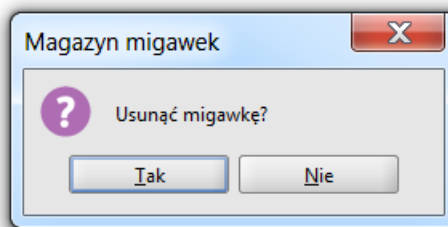
3.1.5 Usuń migawkę

Aby uruchomić funkcję „Usuń migawkę”, należy wybrać:

- **Menu „Migawka”** →  **Usuń migawkę**
- lub
- Ikonę:  .

Funkcja ta:

- pozwala na trwałe usunięcie zaznaczonej (podświetlonej) migawki,
- uaktywnia się dopiero po wybraniu (zaznaczeniu) na liście wymaganej migawki,
- wyświetla komunikat z zapytaniem (zob. Rysunek 89):




Rysunek 89. Zapytanie o usunięcie migawki

Po udzieleniu pozytywnej odpowiedzi:

- migawka znika z magazynu migawek i
- nie jest już widoczna na prezentowanej liście istniejących migawek.

3.1.6 Importuj migawkę

Aby uruchomić funkcję „Importuj migawkę”, należy wybrać:

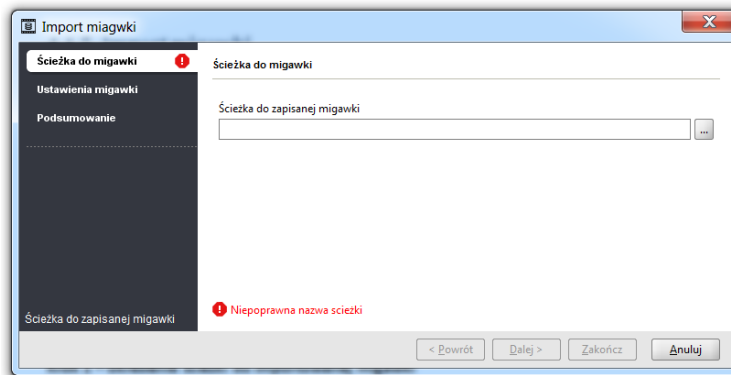
- **Menu „Migawka”** →  **Importuj migawkę**
- lub
- ikonę: 
- lub
- skrót klawiszowy **Ctrl-I** .

Funkcja ta:

- pozwala na zaimportowanie migawki, zapisanej w formacie xml.

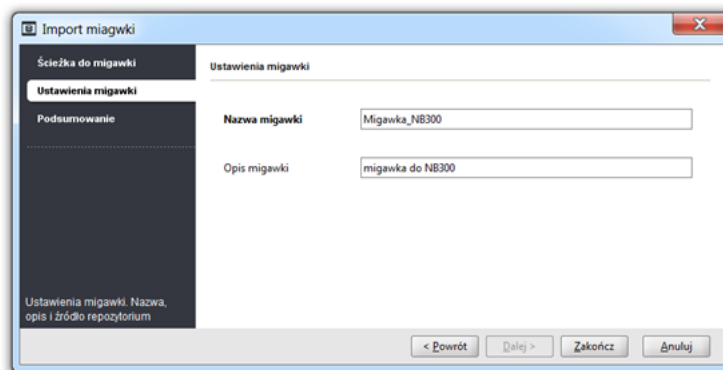
Pełny proces importu migawki wykonywany jest w trzech krokach.

Krok 1 – to określenie ścieżki do importowanej migawki (zob. Rysunek 90).



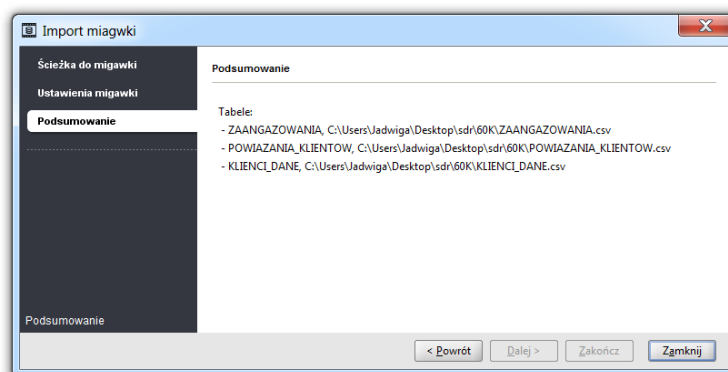
Rysunek 90. Wskazanie ścieżki do importowanej migawki

Krok 2 – to prezentacja ustawień importowanej migawki (zob. Rysunek 91).



Rysunek 91. Prezentacja ustawień importowanej migawki



Krok 3 – to informacja o tabelarycznych źródłach danych, powiązanych z importowaną migawką (zob. Rysunek 92)



Rysunek 92. Informacja o tabelach powiązanych z importowaną migawką

3.1.7 Eksportuj migawkę

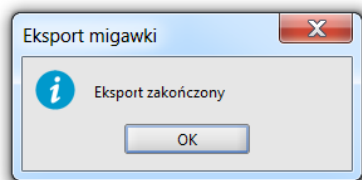
Aby uruchomić funkcję „Eksportuj migawkę”, należy wybrać:

- **Menu „Migawka”** →  Eksportuj migawkę
- lub
- ikonę: 
- bądź
- skrót klawiszowy Ctrl+Shift+E.

Funkcja ta:

- pozwala na wyeksportowanie w formacie xml dowolnej, wcześniej zdefiniowanej migawki;
- uaktywnia się dopiero po wybraniu (zaznaczeniu) na liście migawek tej, dla której ma zostać wykonany eksport;
- po jej uruchomieniu, konieczne jest wskazanie miejsca zapisu wyeksportowanej migawki.



Prawidłowo zakończony proces eksportu sygnalizowany jest w formie komunikatu (zob. Rysunek 93).



Rysunek 93. Informacja o zakończeniu eksportu migawki

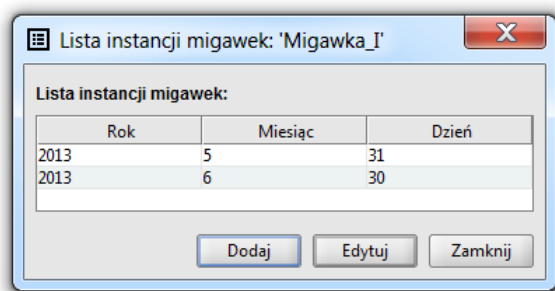
3.1.8 Zarządzaj instancjami migawek

Aby uruchomić funkcję „Zarządzaj instancjami migawek”, należy wybrać:

- **Menu „Migawka”** →  Zarządzaj instancjami migawek
- lub
- ikonę: 
- bądź
- skrót klawiszowy Ctrl+Shift+Z.

Funkcje tu pogrupowane pozwalają na:

- prezentację listy instancji migawek, powiązanych z wcześniej wybraną migawką (zob. Rysunek 94),
- edycję instancji prezentowanych na liście,
- definiowanie nowych instancji migawek.



Rysunek 94. Lista instancji migawek

Funkcje te:

- mogą zostać uaktywnione tylko po wcześniejszym zaznaczeniu na liście migawki, dla której chcemy zarządzać jej instancjami.

Instancje migawek:

- **tworzą się automatycznie** i są powiązane z konkretnymi okresami sprawozdawczymi, gdy przy wykorzystaniu migawkowych źródeł danych wykonane zostaną funkcje:
 - „Zasilenia danymi” (zob. rozdział [Zasilanie według mapowań](#)),
 - „Test źródła danych” (zob. rozdział [Dane podstawowe nowego źródła danych](#)),
- **mogą zostać zdefiniowane ręcznie**, przy wykorzystaniu funkcji:
 - „Dodaj nową instancję migawek” (zob. rozdział [Dodaj nową instancję migawek](#)), ale wówczas nie są powiązane z żadnymi danymi,
- **mogą zostać zasilone danymi**, przy wykorzystaniu funkcji:
 - „Zasilanie według mapowań” (zob. rozdział [Zasilanie według mapowań](#)),
 - „Test źródła danych” (zob. rozdział [Dane podstawowe nowego źródła danych](#)),
 - „Wczytaj dane źródłowe dla zaznaczonych tabel” (zob. rozdział [Wczytaj dane źródłowe dla zaznaczonych tabel](#)).

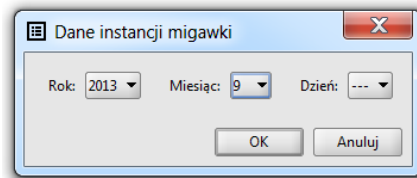
3.1.8.1 Dodaj nową instancję migawek

Aby dla wybranej migawki zdefiniować dodatkową instancję migawek, należy w „Zarządzaniu instancjami migawek”, wybrać:

- przycisk funkcyjny:  .

Funkcja ta:

- pozwala na utworzenie nowej instancji migawek (zob. Rysunek 95),
- wymaga podania okresu, dla którego ma zostać zdefiniowana nowa instancja migawki.



Rysunek 95. Dane instancji migawki

Utworzona instancja migawki:

- nie jest powiązana z żadnymi danymi, ale można do niej;
- zaimportować dane,
- wykonać testowe zapytania SQL.

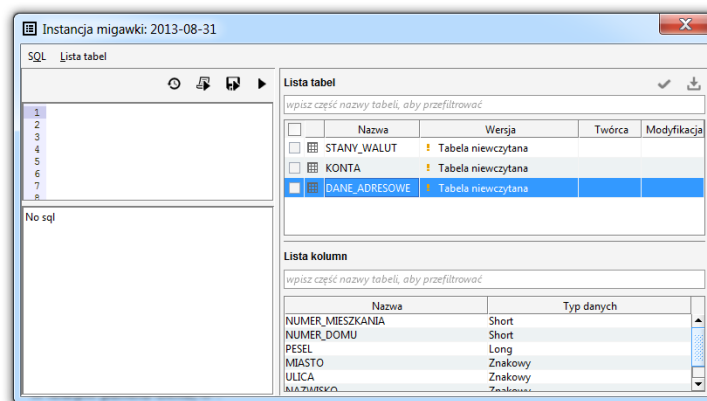
3.1.8.2 Edytuj instancje migawek

Prezentowane na liście instancje migawek mogą podlegać weryfikacji przy wykorzystaniu funkcji „Edytuj instancje migawek”, wybierając:

- przycisk funkcyjny:  .

Funkcja ta pozwala na (zob Rysunek 96):

- załadowanie danymi źródłowymi wybranych tabel powiązanych z migawką,
- wykonanie wszystkich kroków ETL,
- wykonanie wstępnej walidacji poprawności danych w wybranych tabelach,
- wprowadzenie i wykonanie zapytań języka SQL.



Rysunek 96. Informacje szczegółowe prezentowane w instancji migawki

W lewym panelu okna:

- w górnej części – można odwołując się do tabel prezentowanych w prawym panelu okna, zdefiniować dowolne zapytania języka SQL operujące na danych z tabel migawki oraz danych z tabel kroków ETL;

- w dolnej części wyświetlane są wyniki wykonanego zapytania SQL.

W prawym panelu okna, w:

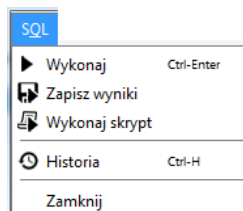
- górnej części wyświetlana jest lista tabel powiązanych z migawką, z uszczegółowieniem informacji do:
 - nazwy tabeli,
 - wersji definicji migawki / ETL połączonej z wersją danych zawartych w instancji,
 - loginu twórcy tabeli,
 - daty modyfikacji,przed nazwą tabeli prezentowany jest kwadrat:
 - pusty, gdy wykonywane funkcje nie dotyczą danej tabeli,
 - zaznaczony, gdy dla danej tabeli ma być wykonana jedna z funkcji:
 - walidacja danych lub
 - wczytanie danych źródłowych,
- dolnej części wyświetlana jest lista kolumn dla wybranej w górnym panelu okna tabeli, z informacjami o:
 - nazwie kolumny,
 - typie danych.

Aby zaznaczyć /odznaczyć wszystkie tabele prezentowane w prawym panelu okna, należy:

- zaznaczyć / odznaczyć kwadra - znajdujący nad prezentowaną listą tabel, w wierszu nagłówkowym górnej części prawego panelu okna.

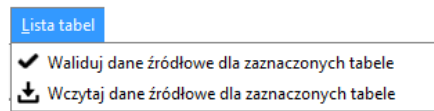
Można tu wykonać wiele funkcji, pogrupowanych w menu:

- SQL (zob. Rysunek 97):
 - Wykonaj,
 - Zapisz wyniki,
 - Wykonaj skrypt,
 - Historia,
 - Zamknij.









Rysunek 97. Menu "SQL" w zarządzaniu instancjami migawek

- Lista tabel (zob. Rysunek 98):
 - Waliduj dane źródłowe dla zaznaczonych tabel,
 - Wczytaj dane źródłowe dla zaznaczonych tabel.



Rysunek 98. Menu "Lista tabel" w zarządzaniu instancjami migawek

Dodatkowo funkcje te, dostępne są również w formie ikon funkcyjnych, znajdujących się poniżej menu funkcyjnego:

- w lewym panelu okna:
 -  - historia,
 -  - wykonaj skrypt,
 -  - zapisz wyniki,
 -  - wykonaj,
- w prawym panelu okna:
 -  - waliduj dane źródłowe dla zaznaczonych tabel,
 -  - wczytaj dane źródłowe dla zaznaczonych tabel.

3.1.8.2.1 Wykonywanie skryptów SQL

W celach testowych i weryfikacji danych źródłowych powiązanych z utworzonymi instancjami migawek, możliwe jest definiowanie i wykonywanie zapytań SQL, odnoszących się do tabel zdefiniowanych migawek.

W menu „SQL” prezentowanej instancji migawki dostępne są dwie funkcje, pozwalające na wykonanie zdefiniowanych zapytań:

- funkcja „Wykonaj” – dotycząca zapytań wprowadzonych przez samego użytkownika, do górnego lewego panelu okna;
- funkcja „Wykonaj skrypt” – dotycząca skryptów dostarczonych przez dostawcę aplikacji aSIS, w formie plików zip.

Aby wykonać skrypt zdefiniowany przez użytkownika, odnoszący się do tabel instancji migawek, należy w „Zarządzaniu instancjami migawek” będąc w edycji wybranej instancji:

- do lewego górnego panelu okna wprowadzić zapytania SQL,

a następnie wybrać:

- **Menu „SQL”** →  Wykonaj

lub

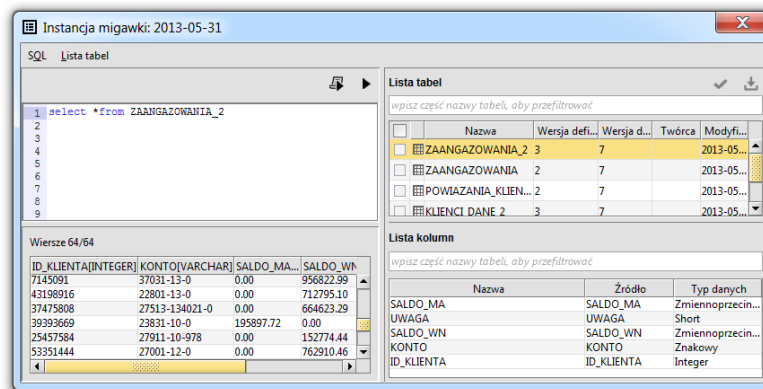
- Ikonę: 

lub

- skrót klawiszowy: Ctrl-Enter.

Funkcja ta:

- w dolnym panelu lewego okna zaprezentuje:
 - wynik prawidłowo zdefiniowanego zapytania SQL (Rysunek 99) lub
 - informacje o błędach w sformułowanym zapytaniu,
- może wykonać się prawidłowo tylko dla tabel, posiadających wczytane dane źródłowe.



Rysunek 99. Informacje prezentowane w instancji migawki po wykonaniu zapytania SQL

Przy definiowaniu zapytania SQL pomocna może być funkcja pozwalająca na proste i szybkie przeniesienie nazwy tabeli bądź kolumny do definiowanego zapytania, poprzez:

- dwukrotne kliknięcie lewym przyciskiem myszki na wymaganą tabelę czy kolumnę z prawego panelu okna.

W sytuacji, gdy wymagana jest pomoc serwisu aSIS w zdefiniowaniu skomplikowanych zapytań SQL, funkcją umożliwiającą wykonanie dostarczonych skryptów jest funkcja: „Wykonaj skrypt” wykonana w „Zarządzaniu instancjami migawek”, w edycji wybranej instancji, poprzez wybranie:

- **Menu „SQL”** →  Wykonaj skrypt

lub



- Ikony:  .

Funkcja ta:

- wymaga wskazania pliku ze skryptami SQL, zapisanymi jako *.zip;
- w dolnym panelu lewego okna zaprezentuje wynik zdefiniowanego zapytania SQL.

3.1.8.2 Zapisanie wyników zapytania SQL

Aby zapisać otrzymany wynik zdefiniowanego zapytania SQL, należy uruchomić funkcję „Zapisz wyniki”, poprzez wybranie:



- Menu „SQL” →  Zapisz wyniki
- lub
- Ikony:  .

Funkcja ta:

- wymaga wskazania miejsca do zapisu pliku ze skryptami SQL, zapisanymi jako *.zip

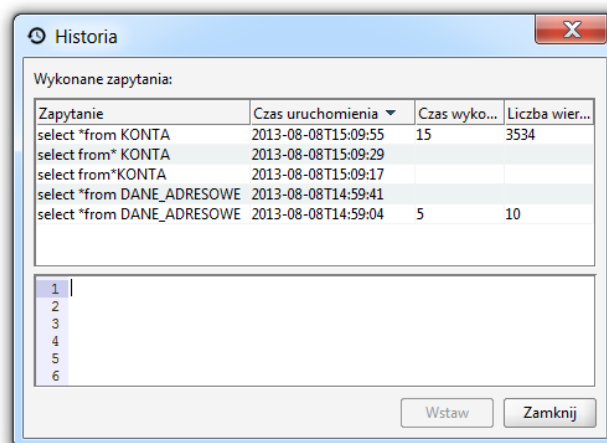
3.1.8.3 Historia wykonanych zapytań SQL

Aby sprawdzić historię wykonanych zapytań SQL dla wybranej instancji migawki, należy uruchomić funkcję „Historia zapytań SQL”, wybierając:

- Menu „SQL” →  Historia
- lub
- Ikonę: 
- bądź
- skrót klawiszowy: Ctrl-H.

Funkcja ta:



- w nowym oknie (zob. Rysunek 100), prezentuje wszystkie zapytania SQL, wykonane dla danej instancji migawki;
- dodatkowo informuje o:
 - terminie wykonania zapytania SQL,
 - czasie wykonywania zapytania,
 - liczbie wierszy biorących udział w wykonywanym zapytaniu SQL.




Rysunek 100. Historia zapytań SQL

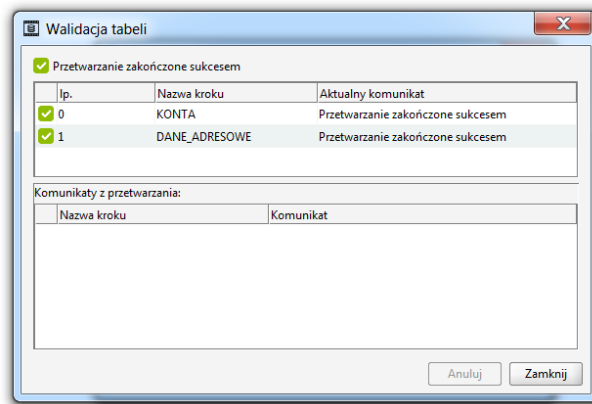
3.1.8.2.4 Walidacja danych w instancji migawki

Aby uruchomić walidację danych źródłowych powiązanych z instancją migawek, należy w „Zarządzaniu instancjami migawek” będąc w edycji wybranej instancji, uruchomić funkcję „Waliduj dane źródłowe dla zaznaczonych tabel”, wybierając:

- Menu „Lista tabel” →  Waliduj dane źródłowe dla zaznaczonych tabel
- lub
- Ikonę:  .

Funkcja ta:



- uaktywnia się dopiero po wcześniejszym zaznaczeniu tabel, dla których na zostać wykonana walidacja (zaznaczony kwadrat );
- informuje o wynikach przeprowadzonej walidacji (zob. Rysunek 101),
- po wystąpieniu błędów walidacyjnych, informuje o powstałych błędach.




Rysunek 101. Informacja o wynikach przeprowadzonej walidacji danych w instancji migawek

3.1.8.2.5 Wczytaj dane źródłowe dla zaznaczonych tabel

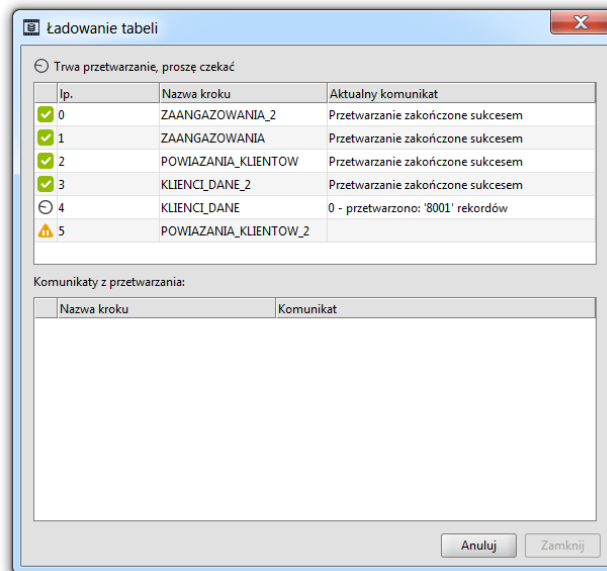
Aby zaimportować dane do wybranych tabel powiązanych z instancją migawek lub też rozpocząć przetwarzanie ETL, należy w „Zarządzaniu instancjami migawek” będąc w edycji wybranej instancji, uruchomić funkcję „Wczytaj dane źródłowe dla zaznaczonych tabel”, wybierając:

- Menu „Lista tabel” →  Wczytaj dane źródłowe dla zaznaczonych tabel
- lub
- Ikonę:  .

Funkcja ta:

- uaktywnia się dopiero po wcześniejszym zaznaczeniu tabel, dla których mają zostać zaimportowane dane (zaznaczony kwadrat ) lub też opcjonalnego kroku ETL;
- wczytuje dane do zaznaczonych tabel,

- prezentuje przebieg i wynik procesu importu (zob. Rysunek 102):

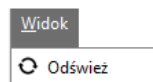


Rysunek 102. Przebieg procesu importu danych do instancji migawek

3.2 Widok

Menu „**Widok**” (zob. Rysunek 103), udostępnia funkcję związaną z:

- odświeżaniem informacji prezentowanych na ekranie.



Rysunek 103. Menu "Widok" w module „Magazyn Migawek”

3.2.1 Odśwież

Aby odświeżyć informacje prezentowane na ekranie, należy wybrać funkcję: „Odśwież”, wybierając:

- **Menu Widok** →  **Odśwież**
- lub
- ikonę .

4 ETL

ETL (ang. *Extract, Transform and Load*) – to narzędzie wspomagające proces transformacji danych zawartych w tabelach migawki.

Jego zadaniem jest przekształcenie danych źródłowych, poprzez:

- łączenie danych (np. łączenie tabeli klientów z ich transakcjami),
- filtrowanie danych (wykonanie odpowiedniego filtrowania danych zgodnie z kryteriami oczekiwanymi przez nadzorcę),
- weryfikację danych – wprowadzenie walidatorów pozwalających na weryfikację spójności i poprawności danych wygenerowanych z systemów backoffice.

Do definiowania przekształceń strukturalnych ETL używany jest standard **SQL**. Dane przechowywane są w bazie **Derby**. Zakres wykorzystania zapytań bazodanowych w **ETL** jest powiązany z funkcjonalnościami zapewnionymi przez dostawcę bazy danych.

Szczegółowy opis tworzenia zapytań znajduje się w dokumencie: **SQL language reference** na stronie: <https://db.apache.org/derby/docs/10.11/ref/crefsqj23296.html>.

Niektóre typy przekształceń opierają się na skryptach języka **Groovy**.

Opis języka dostępny jest na stronie: <http://www.groovy-lang.org/documentation.html> w rozdziałach: *Syntax, Operators, Semantics*.

W **ETL** stosowane są skrypty w ograniczonym kontekście funkcjonalnym, dlatego też warto zapoznać się z przykładami przedstawionymi w niniejszym dokumencie.

4.1 Kroki ETL

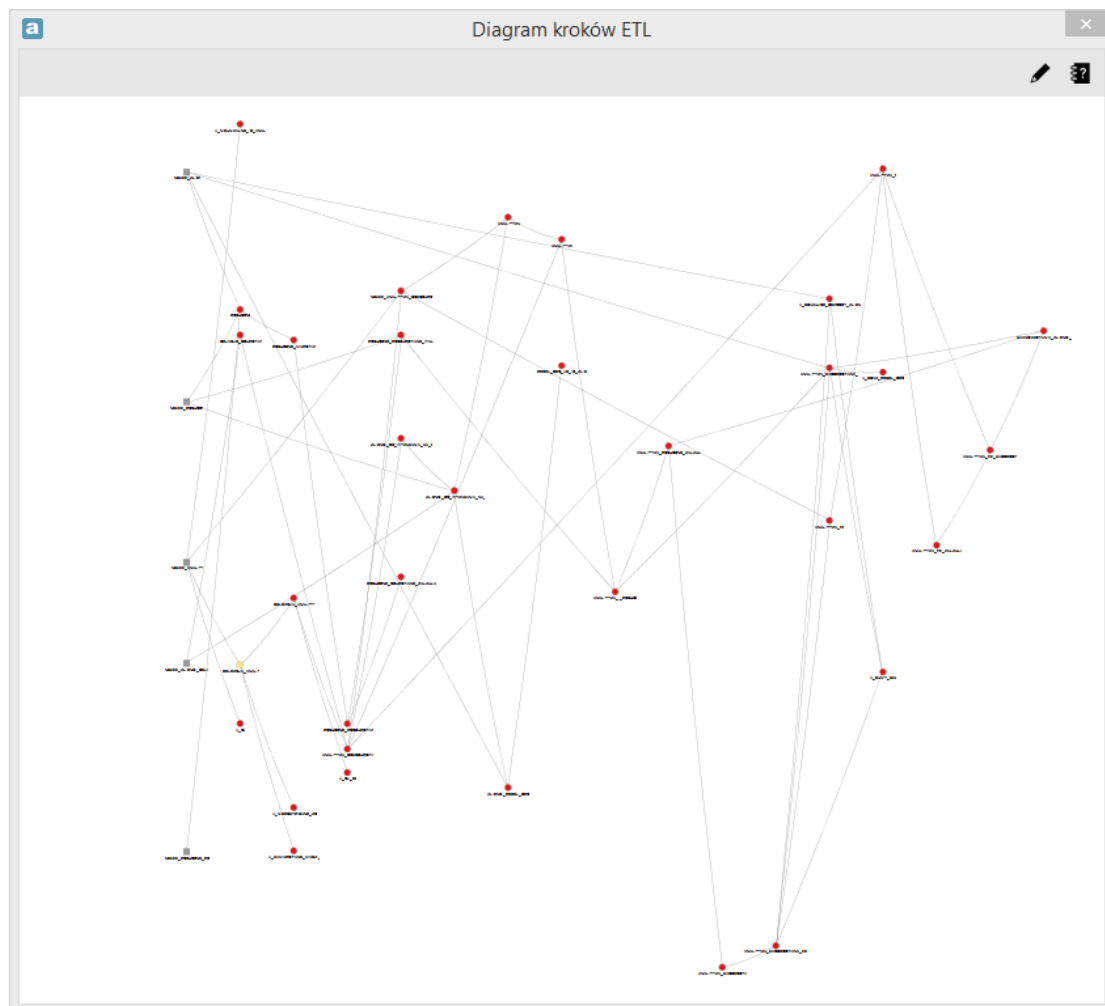
Podstawowym elementem przetwarzania danych zaimplementowanym w ETL jest:

- **krok**

Każdy krok korzysta z co najmniej jednej tabeli relacyjnej bazy danych. Dane wejściowe nie są modyfikowane. Wyniki przetwarzania są zapisywane w jednej tabeli wyjściowej.

Takie rozwiązanie pozwala na traktowanie kroku jako tabelę wejściową innych kroków. Kolejne definicje tworzą zależności pomiędzy krokami wymuszając sekwencyjne wykonywanie operacji.

Rysunek 104 przedstawia przykład wizualizacji zależności jakie mogą powstać podczas pracy z ETL.



Rysunek 104. Diagram kroków ETL

Przy wykonywaniu wielu kroków, wystąpienie błędów podczas przetwarzania jednego z nich, przerywa dalsze wykonanie kroków danej sekwencji. Przetwarzanie danych nienależących do sekwencji przerwanej jest kontynuowane.



Wykonanie kroku bywa nazywane:

- zasileniem danymi
- lub:
- wczytaniem tabeli.

4.1.1 Krok wejściowy

Aby uruchomić krok wejściowy w ETL, należy wybrać:

- **Menu „Migawka”** →  Edytuj ETL
- lub

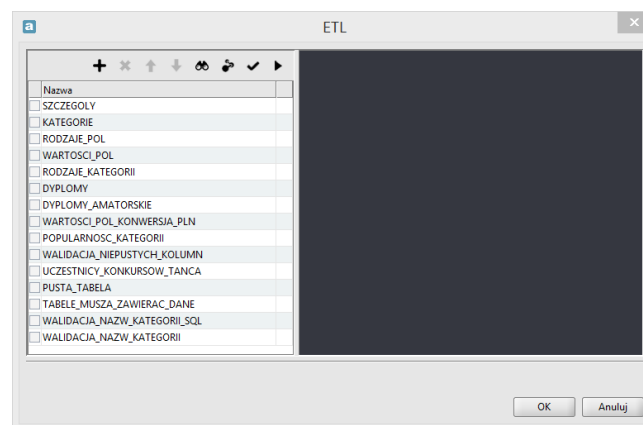
- ikonę: 
 - i opcję:  Edytuj ETL
- bądź

- skrót klawiszowy Ctrl+Shift+W.

Każdorazowo konieczne jest wcześniejsze:

- zaznaczenie (podświetlenie) migawki, dla której będą wykonywane przekształcenia tabel migawkowych.

Po uaktywnieniu kroku wejściowego, prezentowany jest ekran (zob. Rysunek 105):



Rysunek 105. ETL – krok wejściowy

W górnej części lewego panelu okna:

- dostępne są funkcje pozwalające na:
 - odpowiednie przefiltrowanie danych w tabelach,
 - wykonanie skryptów SQL i GROOVY,
 - walidację danych poza pomocą skryptów SQL i GROOVY,
 - usunięcie błędnych lub niepotrzebnie utworzonych kroków,
 - zmianę kolejności prezentowania w oknie utworzonych kroków,
 - wstępną walidację utworzonych kroków,
 - testowe wykonanie zapytania SQL,
- wyświetlana jest lista:
 - tabel zdefiniowanych w wybranej migawce,
 - tabel (kroków) utworzonych w wyniku wykonanych przekształceń.

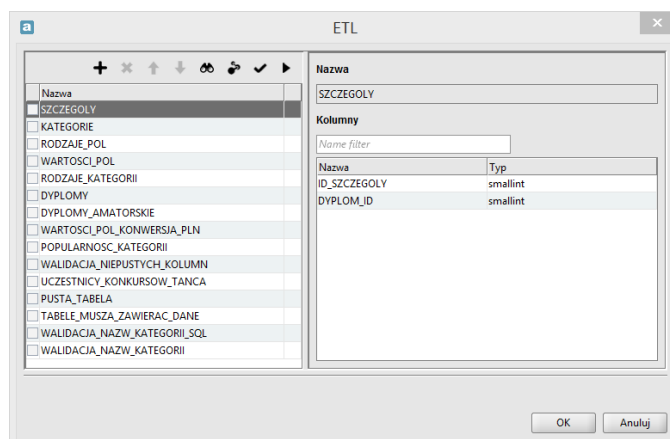
W górnej części prawego panelu okna, po wybraniu (zaznaczeniu) jednej z tabel prezentowych w lewym górnym panelu okna:

- w przypadku tabel migawkowych, wyświetlana jest lista kolumn, z których złożona jest dana tabela (zob. Rysunek 105);

- dla tabel (kroków) utworzonych w wyniku uaktywnienia dostępnych funkcji, prezentowane są dodatkowe funkcje umożliwiające wykonanie stosownych przekształceń tabel migawkowych (zob. Rysunek 106).

W dolnej części okna, wyświetlane są:

- wyniki wykonanych skryptów SQL czy GROOVY,
- powstałe błędy.




Rysunek 106. ETL – krok wejściowy z widocznym opisem tabeli źródłowej

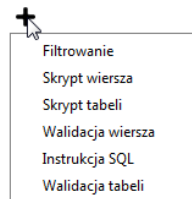
Niektóre z dostępnych tu funkcji mogą być zawsze wykorzystywane a inne uaktywniają się dopiero po zaznaczeniu kwadratu występującego przed nazwą tabeli (kroku), której mają dotyczyć (zob. Rysunek 106).

Dane są przechowywane w tabelach relacyjnej bazy danych SDR (ang. Source Data Repository – repozytorium danych źródłowych). Tabele te są dostępne do dalszego przetwarzania w ETL. Ich cechą charakterystyczną jest to, że nie mają one własnych kroków wejściowych. Definicja migawki określa jakie tabele z danymi źródłowymi są dostępne w ETL.

4.2 Tworzenie nowego kroku

Aby utworzyć “Nowy krok”, należy wybrać (zob. Rysunek 107):

- ikonę: ,
- a następnie jedną z dostępnych funkcji:
 - Filtrowanie,
 - Skrypt wiersza,
 - Skrypt tabeli,
 - Walidacja wiersza,
 - Instrukcja SQL,
 - Walidacja tabeli,




Rysunek 107. ETL – funkcje tworzące nowy krok

Wybranie dowolnej funkcji, każdorazowo powoduje:

- utworzenie nowego kroku i prezentację go w lewym panelu;
- prezentację w prawym panelu okna:
 - nazwy utworzonego kroku,
 - nazwy tabeli migawkowej, dla której wykonywana jest funkcja;
 - wykorzystywane kolumny i indeksy;
 - pola, do którego wprowadzane są stosowne polecenia;
- udostępnienie funkcji pozwalających na odpowiednie przekształcenie tabel migawkowych.

Dla utworzonych kroków, funkcje ułatwiające wykonanie przekształceń tabel, udostępnione zostały w formie ikon funkcyjnych jakich jak:

 – dodaj krok wejściowy / kolumnę / walidator,

 – edytuj kolumnę /skrypt /walidator,

 – kopiuj kolumnę,

 – duplikuj kolumnę,


 – usuń krok wejściowy/ kolumnę / indeks /walidator,

 – dodaj indeks.

Poszczególne ikony widoczne są na rysunkach prezentujących ekrany po uruchomieniu poszczególnych funkcji tworzących nowe kroki do przekształceń tabel.

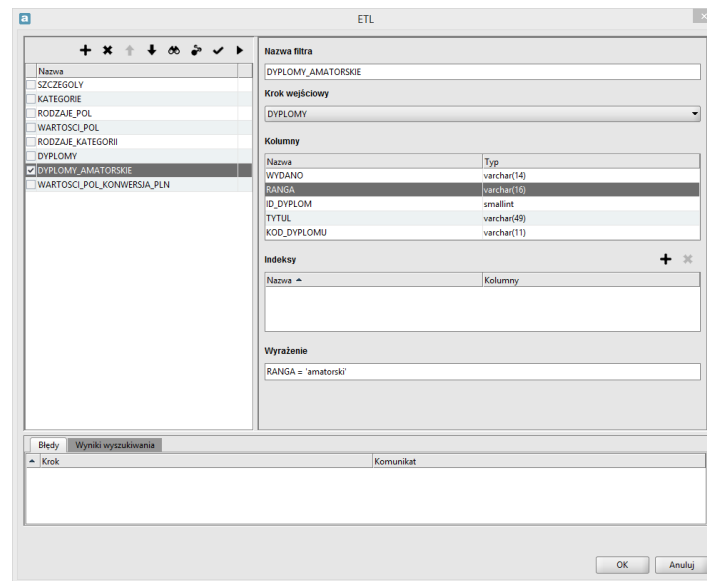
4.2.1 Filtrowanie

Wybranie funkcji „Filtrowanie” poprzez:

- Ikonę:  ,
- a następnie:
- Filtrowanie,

powoduje:

- wybranie wierszy z tabeli źródłowej, spełniających określony warunek;
- prezentację w prawym panelu okna odpowiednich funkcji i pól (zob. Rysunek 108).



Rysunek 108. ETL – filtrowanie danych w tabelach migawkowych

Dane źródłowe pochodzą z jednej tabeli. Tabela wynikowa ma taką samą strukturę jak tabela kroku wejściowego i jej kolumny są tworzone automatycznie.

Zdefiniowany krok:

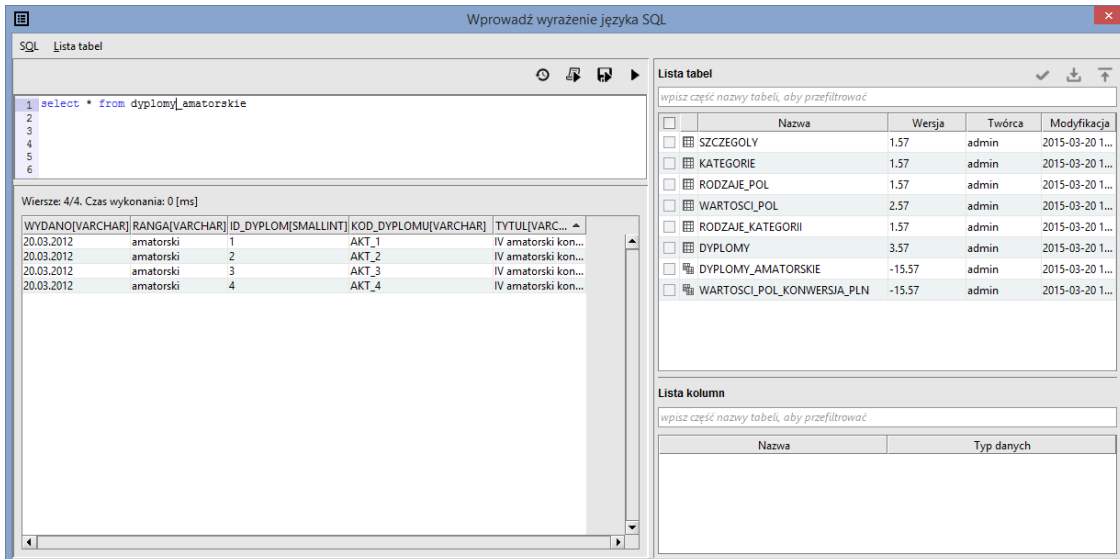
- wybiera z tabeli wejściowej wiersze, które mają określoną wartość w jednej z kolumn.

Warunek filtrujący definiowany jest w polu **Wyrażenie**. Treść warunku jest klauzulą WHERE wyrażenia **SQL**.

Ogólna treść zapytania wykonywanego zapytania:

```
INSERT INTO tabela docelowa ( nazwy kolumn ) SELECT nazwy kolumn FROM tabela kroku wejściowego WHERE wyrażenie
```

Zawartość tabeli po wykorzystaniu funkcji filtrowania (zob. Rysunek 109):



Wiersze: 4/4. Czas wykonania: 0 [ms]

WYDANO[VARCHAR]	RANGA[VARCHAR]	ID_DYPLOM[SMALLINT]	KOD_DYPLOMU[VARCHAR]	TYTUŁ[VARC...
20.03.2012	amatorski	1	AKT_1	IV amatorski kon...
20.03.2012	amatorski	2	AKT_2	IV amatorski kon...
20.03.2012	amatorski	3	AKT_3	IV amatorski kon...
20.03.2012	amatorski	4	AKT_4	IV amatorski kon...

Lista tabel

Nazwa	Wersja	Twórca	Modyfikacja
SZCZEGOLY	1.57	admin	2015-03-20 1...
KATEGORIE	1.57	admin	2015-03-20 1...
RODZAJE_POL	1.57	admin	2015-03-20 1...
WARTOSCI_POL	2.57	admin	2015-03-20 1...
RODZAJE_KATEGORII	1.57	admin	2015-03-20 1...
DYPLOMY	3.57	admin	2015-03-20 1...
DYPLOMY_AMATORSKIE	-15.57	admin	2015-03-20 1...
WARTOSCI_POL_KONWERSJA_PLN	-15.57	admin	2015-03-20 1...

Lista kolumn

Nazwa	Typ danych
-------	------------


Rysunek 109. Wynik filtrowania

4.2.2 Skrypt wiersza

Dane kroku wejściowego można przetwarzać sekwencyjnie wiersz po wierszu. W tym celu, należy:

- zdefiniować krok typu „Skrypt wiersza”.

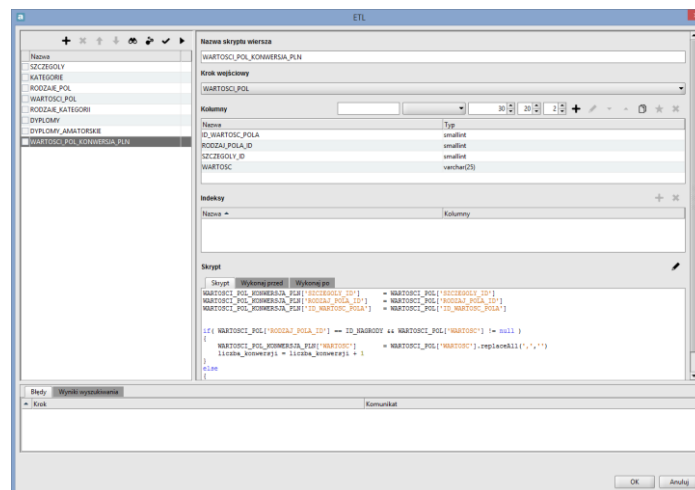
Aby uruchomić tę funkcję, należy wybrać:

- ikonę: ,
- a następnie:

- skrypt wiersza,

spowoduje to:

- utworzenie nowego kroku, pozwalającego na zdefiniowanie skryptu w języku Groovy, który zostanie uruchomiony dla każdego wiersza;
- prezentację w prawym panelu okna odpowiednich pól i funkcji (zob. Rysunek 110).



Rysunek 110. Skrypt wiersza – definicja kroku

Rysunek 110 przedstawia panel do przetwarzania wierszy danych wejściowych. Dane wejściowe są określone przez pojedynczy:

- **krok wejściowy.**

Do określenia tabeli wyjściowej można użyć wybranych kolumn tabeli wejściowej jak również definiować własne.

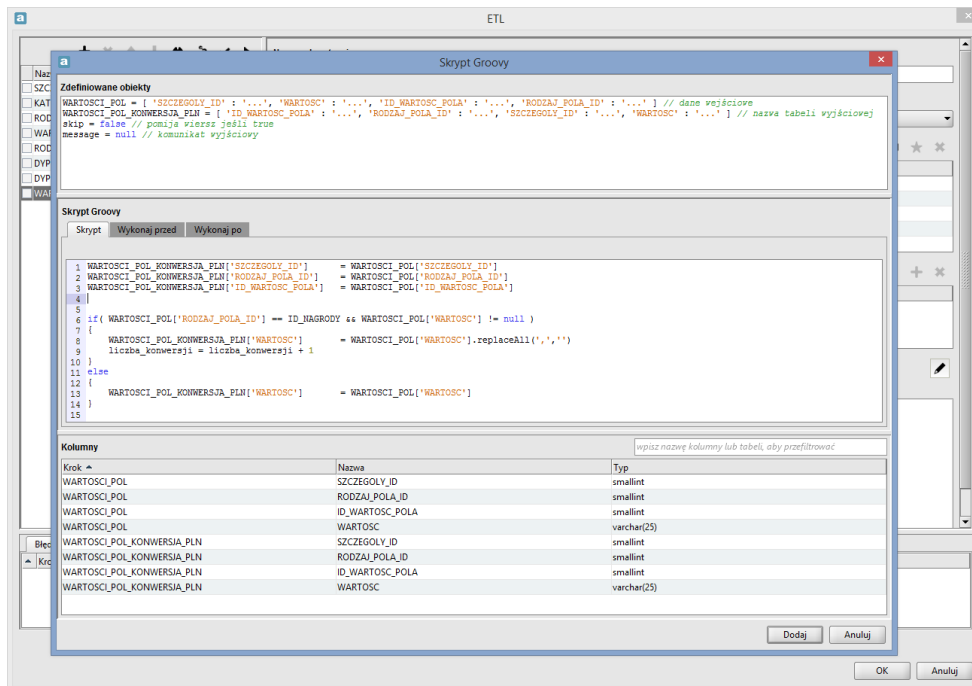
Tabela `WARTOSCI_POL`, danych przykładowych, w kolumnie `WARTOSC` zawiera łańcuchy znaków, których znaczenie jest zdefiniowane poprzez relację z tabel `RODZAJE_POL`.

Przykładowo:

Jednym z rodzajów danych jest wysokość nagród pieniężnych. Kwoty nagród, w danych wejściowych, są zapisane z użyciem przecinka jako separatora tysięcy np. `1,234,567.89`. Skrypt ma za zadanie usunąć separatory z tabel (zapisać kwoty w postaci: `1234567.89`).

W środkowej części panelu edycji skryptu (zob. Rysunek 111) znajdują się zakładki do zdefiniowania trzech głównych faz przetwarzania danych, które będą wykonywane kolejno:

- **wykonaj przed,**
- **skrypt**
 - zostanie uruchomiony wielokrotnie (raz dla każdego wiersza tabeli z danymi wejściowymi), oraz:
- **wykonaj po.**



Rysunek 111. Skrypt wiersza – panel edycji skryptu

4.2.2.1 Wykonaj przed

Funkcja Wykonaj przed służy do:

- definiowania stałych oraz:
- zmiennych dotyczących całej tabeli.

Zadeklarowane zmienne są dostępne podczas dalszego przetwarzania.

```

ID_NAGRODY = 6 // identyfikator nagrody w tabeli RODZAJE_POL (związany z konstrukcją danych wejściowych)
liczba_konwersji = 0 // przechowuje informacje o ilości zmienionych wierszy

```

4.2.2.2 Skrypt

Skrypt korzysta z pojedynczego wiersza tabeli wejściowej i zapisuje jako wiersz tabeli wyjściowej. Dane przechowywane są w mapach o nazwach jak tabele z których dane pochodzą. Kluczami dostępu do danych są nazwy kolumn tabeli.

Mapa tabeli wejściowej służy do:

- odczytu oraz
- zapisu wierszy.

Informacje o powyższych strukturach są zaprezentowane w polach **Zdefiniowane obiekty** oraz **Kolumny**.

Zmienne specjalne:

- `skip` – ustawiona na `true` spowoduje, że:
 - przetwarzanie wiersza nie zakończy się zapisem danych wyjściowych (wiersz zostanie pominięty),
- `message` – jest używana do prezentacji komunikatu po zakończeniu wykonania kroku.

Poniżej podano przykładowy skrypt prezentujący sposób dostępu do danych wiersza.

Warunek `if` sprawdza czy przetwarzany wiersz zawiera informacje o nagrodach pieniężnych. Jeżeli tak, to wówczas zapisuje wartość bez przecinków.

```
WARTOSCI_POL_KONWERSJA_PLN['SZCZEGOLY_ID'] = WARTOSCI_POL['SZCZEGOLY_ID']
WARTOSCI_POL_KONWERSJA_PLN['RODZAJ_POLA_ID'] = WARTOSCI_POL['RODZAJ_POLA_ID']
WARTOSCI_POL_KONWERSJA_PLN['ID_WARTOSC_POLA'] = WARTOSCI_POL['ID_WARTOSC_POLA']

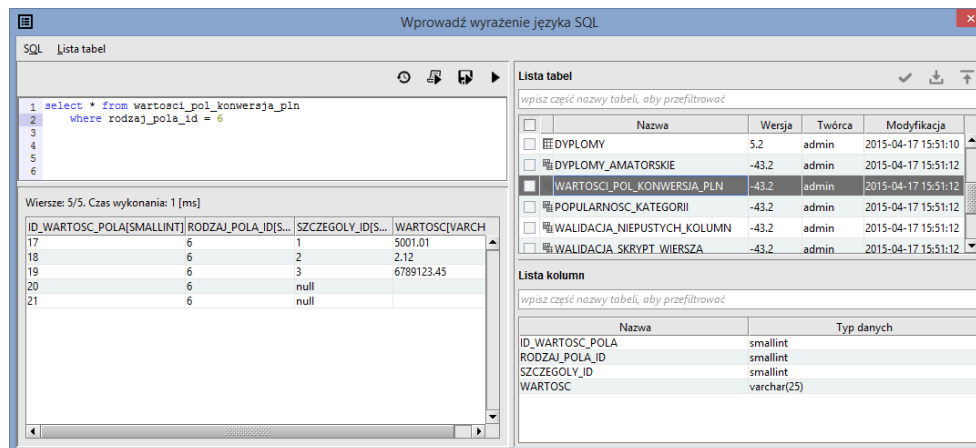
if( WARTOSCI_POL['RODZAJ_POLA_ID'] == ID_NAGRODY && WARTOSCI_POL['WARTOSC'] != null )
{
    WARTOSCI_POL_KONWERSJA_PLN['WARTOSC'] = WARTOSCI_POL['WARTOSC'].replaceAll(',','')
    liczba_konwersji = liczba_konwersji + 1
}
else
{
    WARTOSCI_POL_KONWERSJA_PLN['WARTOSC'] = WARTOSCI_POL['WARTOSC']
}
```

4.2.2.3 Wykonaj po

Wykonaj po zostało użyte do sformułowania komunikatu przedstawionego po zakończeniu kroku.

```
message = 'Dokonano ' + liczba_konwersji + ' konwersji wierszy.'
```


Dane wynikowe zapisane do tabeli kroku (zob. Rysunek 112):



Rysunek 112. Skrypt wiersza – wynik przetwarzania

4.2.3 Skrypt tabeli

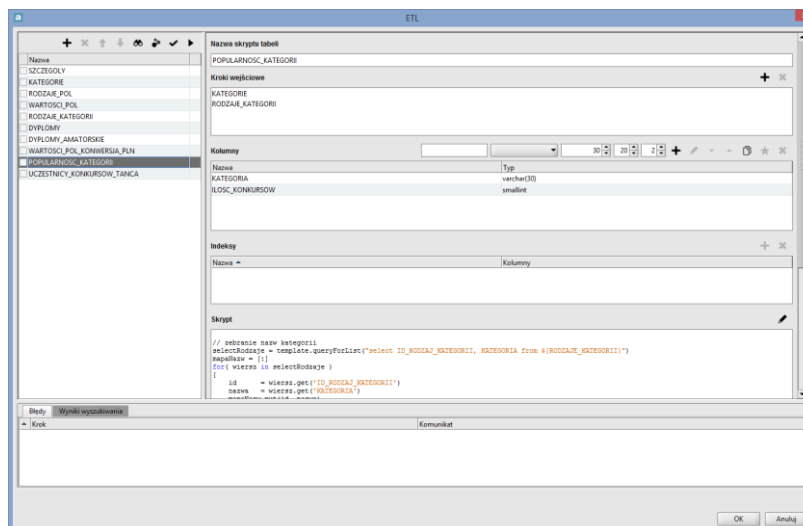
Wybranie funkcji „Skrypt tabeli” poprzez:

- Ikonę: ,
- Skrypt tabeli,

spowoduje:

- utworzenie nowego kroku, pozwalającego na zdefiniowanie skryptu w języku Groovy, w którym można wykonać operacje na całej tabeli za pomocą zapytań SQL.

Operację bazującą na dwóch krokach wejściowych (zob. Rysunek 113):



Rysunek 113. Skrypt tabeli – definicja kroku

Danymi wyjściowymi są informacje statystyczne. Poszukiwane są liczby wydanych dyplomów w każdej z kategorii.

W górnej części panelu do edycji skryptu „Zdefiniowane obiekty” (zob. Rysunek 114):

- znajdują się nazwy z użyciem których można uzyskać dostęp do danych źródłowych oraz:
- widoczna jest nazwa tabeli wyjściowej.

Dozwolonymi operacjami dla:

- tabel wejściowych są:
 - zapytania `select`,
- tabeli wyjściowej jest:
 - komenda `insert`.

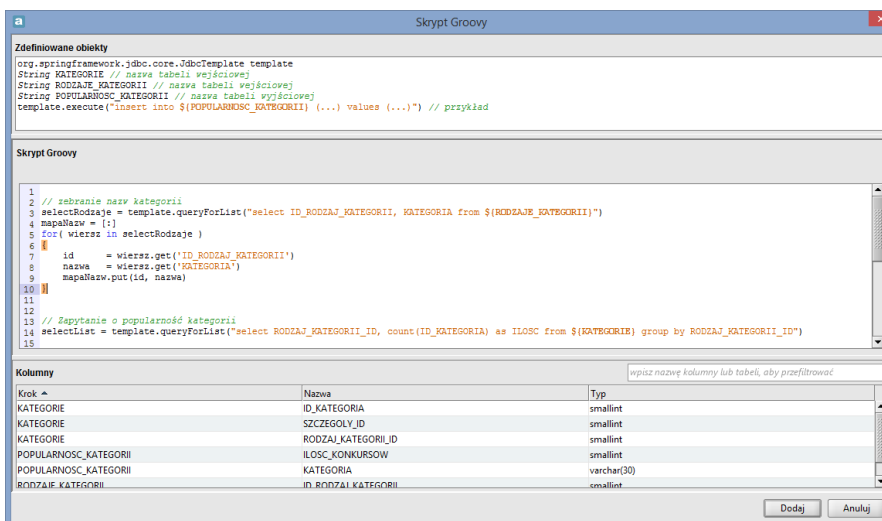
W skryptach tabel zdefiniowane są:

- zapytania do tabel.

Rzeczywiste nazwy tabel w których przechowywane są dane zależą od kontekstu wykonywania skryptu (ten sam skrypt może być wykonany z użyciem różnych migawek danych) i nie są znane podczas jego przygotowywania. Z tego powodu nazwy tabel są traktowane:

- jak parametry (placeholder expression) i należy odnosić się do nich w formie:
 - `${NAZWA_TABELI}`.

W podobny sposób przekazywane są zmienne definiowane w skrypcie do zapytań **SQL**.



The screenshot shows a window titled "Skrypt Groovy" with two main sections. The top section, "Zdefiniowane obiekty", contains Groovy code defining a JdbcTemplate and several strings for table names. The bottom section, "Skrypt Groovy", contains a Groovy script that queries a database for category names and popularity counts. Below the code is a table with columns "Krok", "Nazwa", and "Typ".

```

Zdefiniowane obiekty
org.springframework.jdbc.core.JdbcTemplate template
String KATEGORIE // nazwa tabeli wejściowej
String RODZAJ_KATEGORII // nazwa tabeli wejściowej
String POPULARNOSC_KATEGORII // nazwa tabeli wyjściowej
template.execute("insert into ${POPULARNOSC_KATEGORII} (...) values (...)") // przykład

Skrypt Groovy
1 // zbranie nazw kategorii
2 // selectRodzaje = template.queryForList("select ID_RODZAJ_KATEGORII, KATEGORIA from ${RODZAJ_KATEGORII}")
3 // mapaNazw = []
4 // for( wiersz in selectRodzaje )
5 // {
6 //     id = wiersz.get('ID_RODZAJ_KATEGORII')
7 //     nazwa = wiersz.get('KATEGORIA')
8 //     mapaNazw.put(id, nazwa)
9 // }
10 // zapytanie o popularność kategorii
11 // selectList = template.queryForList("select RODZAJ_KATEGORII_ID, count(ID_KATEGORIA) as ILOSC from ${KATEGORIE} group by RODZAJ_KATEGORII_ID")
12 // }
13 // }
14 // }
15 // }

Kolumny
Krok Nazwa Typ
KATEGORIE ID_KATEGORIA smallint
KATEGORIE SZCZEGOLY_ID smallint
KATEGORIE RODZAJ_KATEGORII_ID smallint
POPULARNOSC_KATEGORII ILOSC_KONKURSOW smallint
POPULARNOSC_KATEGORII KATEGORIA varchar(30)
RODZAJ_KATEGORII ID_RODZAJ_KATEGORII smallint

```

Rysunek 114. Skrypt tabeli - panel edycji skryptu

Dostęp do bazy danych jest realizowany przy użyciu:

- obiektu `template`.

Szczegółowe informacje o możliwych operacjach, związanych z obiektem `template` dostępne są na stronach dokumentacji klasy: `org.springframework.jdbc.core.JdbcTemplate`.

Należy mieć jednak świadomość, że nie wszystkie operacje mogą być dostępne w module **ETL**.

Przykładowo:

Poniżej przedstawiono skrypt przetwarzający dane w trzech fazach.

Początkowo zebrana jest lista kategorii konkursów po czym przygotowywana jest mapa, której kluczami są ich identyfikatory.

```
// zebranie nazw kategorii
selectRodzaje = template.queryForList("select ID_RODZAJ_KATEGORII, KATEGORIA from
${RODZAJE_KATEGORII}")
mapaNazw = [:]
for( wiersz in selectRodzaje )
{
    id      = wiersz.get('ID_RODZAJ_KATEGORII')
    nazwa   = wiersz.get('KATEGORIA')
    mapaNazw.put(id, nazwa)
}
```

W dalszej części, wykonane jest zapytanie o statystyki występowania kategorii. Identyfikator rodzaju kategorii zostaje zastąpiony czytelną nazwą.

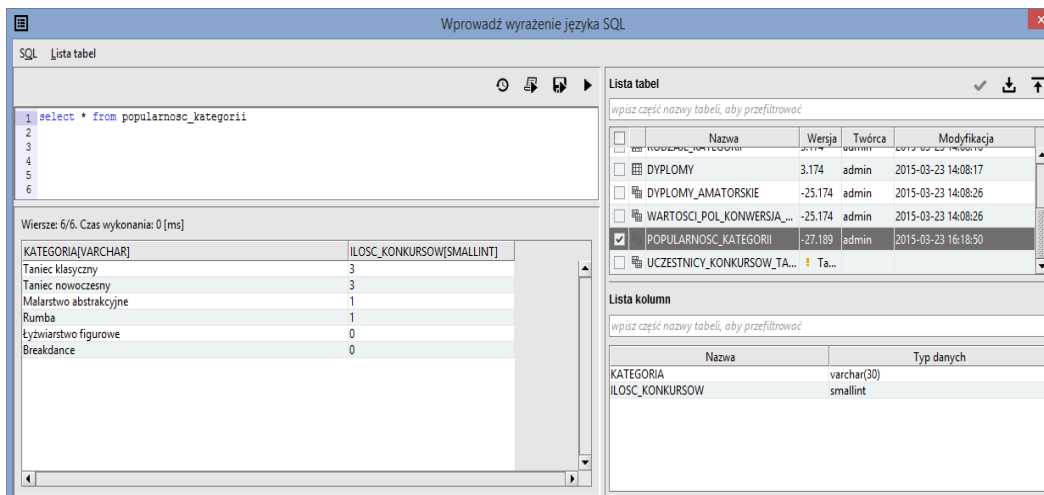
Należy zwrócić uwagę na przekazanie wartości liczbowej `ilosc` bez znaków apostrofu.

```
for( wiersz in selectList )
{
    rodzaj = wiersz.get('RODZAJ_KATEGORII_ID')
    rodzaj = mapaNazw.get(rodzaj)
    ilosc  = wiersz.get('ILOSC')
    template.execute("insert into ${POPULARNOSC_KATEGORII} (KATEGORIA, ILOSC_KONKURSOW)
values ('${rodzaj}', ${ilosc})")
    mapaNazw.remove(wiersz.get('RODZAJ_KATEGORII_ID'))
}
```

Ostatnia część skryptu, zapisuje kategorie w których nie odbyły się konkursy:

```
for( pozostala_kategoria in mapaNazw )
{
    rodzaj = pozostala_kategoria.getValue()
    template.execute("insert into ${POPULARNOSC_KATEGORII} (KATEGORIA, ILOSC_KONKURSOW)
values ('${rodzaj}', 0)")
}
```

Wynik wykonania przykładowego skryptu (zob. Rysunek 115)



Wprowadź wyrażenie języka SQL

SQL Lista tabel

```
1 select * from popularnosc_kategorii
2
3
4
5
6
```

Wiersze: 6/6. Czas wykonania: 0 [ms]

KATEGORIA[VARCHAR]	ILOSC_KONKURSOW[SMALLINT]
Taniec klasyczny	3
Taniec nowoczesny	3
Malarstwo abstrakcyjne	1
Rumba	1
Tyzniarstwo figurowe	0
Breakdance	0

Lista tabel

wpisz część nazwy tabeli, aby przefiltrować

Nazwa	Wersja	Twórca	Modyfikacja
KATEGORIA	3.174	admin	2015-03-23 14:08:17
DYPLOMY	-25.174	admin	2015-03-23 14:08:26
DYPLOMY_AMATORSKIE	-25.174	admin	2015-03-23 14:08:26
WARTOSCI_POL_KONWERSJA...	-25.174	admin	2015-03-23 14:08:26
POPULARNOSC_KATEGORII	-27.189	admin	2015-03-23 16:18:50
UCZESTNICY_KONKURSOW_TA...			

Lista kolumn

wpisz część nazwy tabeli, aby przefiltrować


Nazwa	Typ danych
KATEGORIA	varchar(30)
ILOSC_KONKURSOW	smallint

Rysunek 115. Skrypt tabeli – wynik wykonania

4.2.4 Walidacja wiersza

Moduł ETL umożliwia definiowanie kroków sprawdzających poprawność danych w tabeli poprzez analizę danych wiersz po wierszu.

Wybranie funkcji „Walidacja wiersza” poprzez:

- Ikonę: ,
- a następnie:
 - walidacja wiersza,
- powoduje:
 - utworzenie nowego kroku, pozwalającego na walidację pojedynczych wierszy z danymi za pomocą skryptów Groovy bądź gotowych funkcjonalności.

Krok walidacji wiersza:

- korzysta z pojedynczej tabeli wejściowej,
- może zawierać więcej niż jeden walidator, definiujący poprawność danych.

Jeżeli pojedynczy wiersz zostanie zgłoszony jako błędny przez wiele walidatorów, to wówczas:

- zapisany on zostanie w tabeli wyjściowej tylko raz,
- komentarze zostaną zebrane i zapisane w jednym polu kolumny `comment`.

Wartości błędne wierszy w kroku walidacji, przepisywane automatycznie są do tabeli wyjściowej

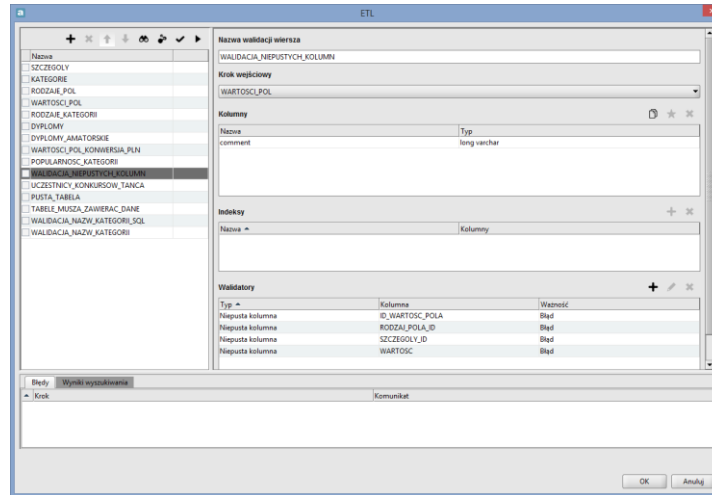
- która posiada strukturę tabeli wejściowej z dodatkową kolumną `comment` przeznaczoną do opisu problemu.

Tabela wyjściowa nie musi zawierać wszystkich kolumn tabeli wejściowej.

4.2.4.1 Niepuste kolumny

Rysunek 116 przedstawia krok walidacji z dodanymi **walidatorami** typu: **niepusta kolumna**.

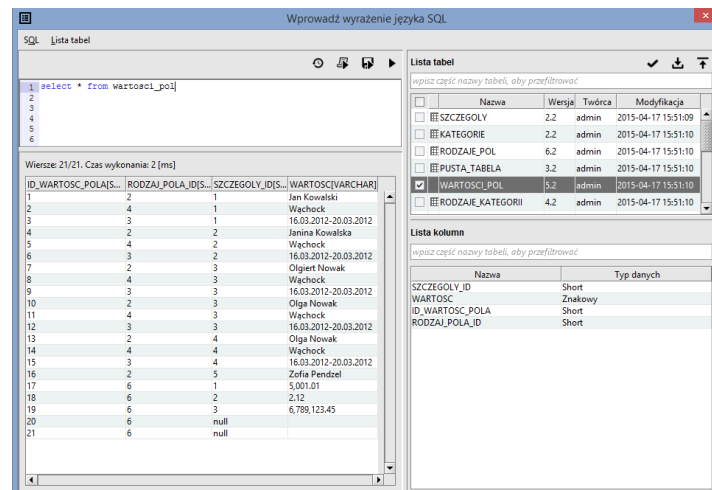
Każdy z walidatorów sprawdza czy w danej kolumnie znajdują się wartości: **null**.



Rysunek 116. Walidacja wiersza, kolumny – definicja kroku

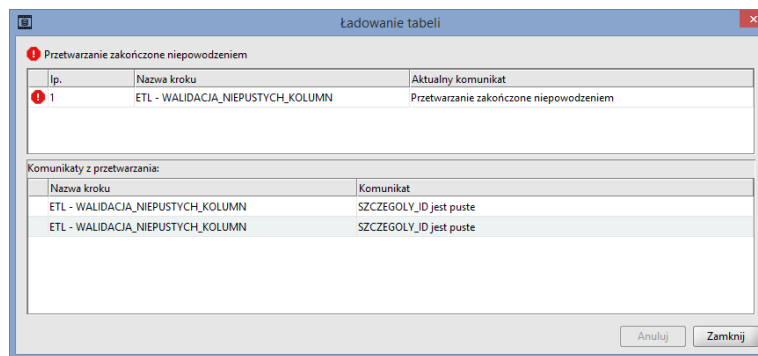
Rysunek 117 i Rysunek 118 przedstawia przykładową sytuację, gdy:

- nie są raportowane „puste” dane niebędące **null**.



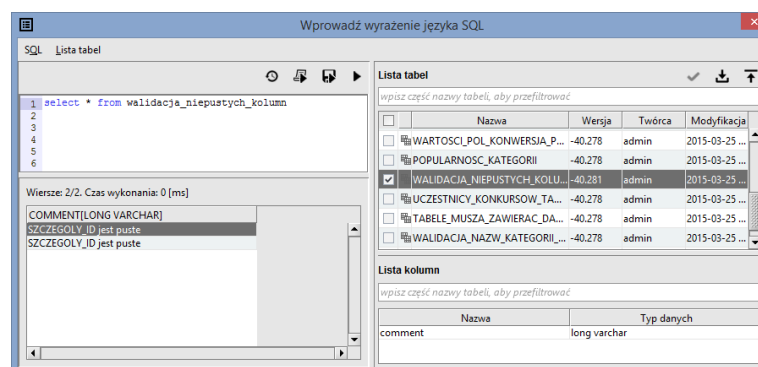
Rysunek 117. Walidacja wiersza, kolumny – zawartość tabeli wejściowej

Przedstawiona sytuacja jest wynikiem algorytmu zasilania danych z plików CSV. Brak danych liczbowych jest oznaczany wartością **null**, natomiast brak danych będącymi łańcuchami znaków jest oznaczany pustym łańcuchem.



Rysunek 118. Walidacja wiersza, kolumny – raport wykonania

Zawartość tabeli wyjściowej po wykonaniu kroku przedstawia Rysunek 119:



Rysunek 119. Walidacja wiersza, kolumny – wynik wykonania

4.2.4.2 Skrypt walidacji wiersza

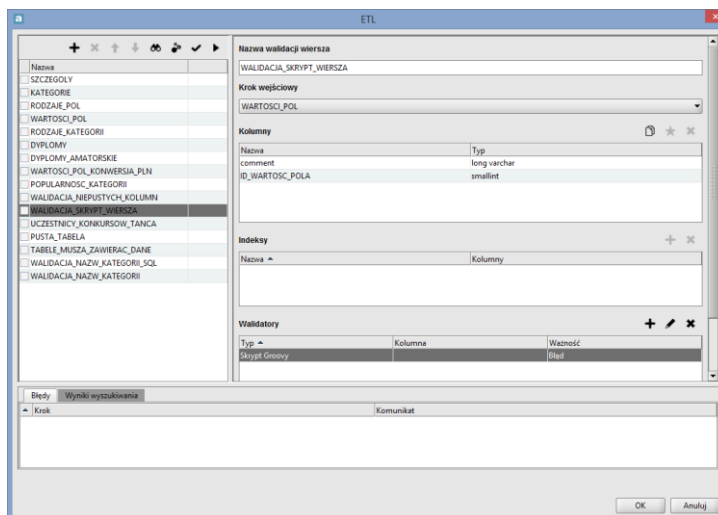
Sprawdzenie wartości wiersza, można przeprowadzić za pomocą:

- walidatora typu skrypt Groovy
 - umożliwiającego dostęp do danych kroku wejściowego.

Przykładowo:

Walidator **niepusta kolumna** sprawdza czy wiersz zawiera wartości `nu11`. Wykrycie pustych łańcuchów znaków można dokonać za pomocą skryptu.

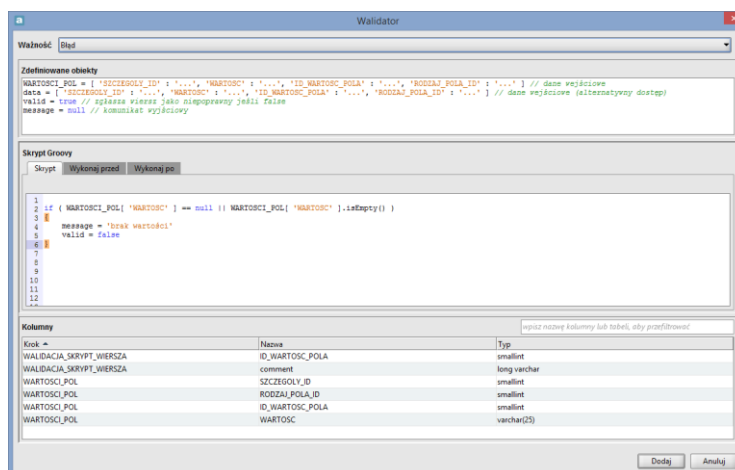
Rysunek 120 przedstawia krok walidujący za pomocą skryptu, gdzie sprawdzana jest tabela (`WARTOSCI_POL`).



Rysunek 120. Walidacja wiersza skryptem - definicja kroku

Tabela wyjściowa zawiera komentarz (`comment`) oraz identyfikator wiersza, w którym znaleziono błąd (`ID_WARTOSC_POLA`).

Implementacja sprawdzenia wartości wierszy przedstawia Rysunek 121:



Rysunek 121. Walidacja wiersza skryptem - panel edycji

Część **zdefiniowane obiekty** (zob. Rysunek 121), obrazuje:

- sposób dostępu do danych.

Mapa wartości `data` jest tożsama z mapą nazwaną nazwą tabeli wejściowej (`WARTOSCI_POL`).

Ze względu na czytelność skryptu, zalecane jest odnoszenie się do danych za pomocą tylko jednej z map.

Do zgłaszania błędów wykorzystuje się zmienne specjalne:

- `valid` – jeżeli skrypt przypisze zmiennej wartość `false`,
 - wiersz zostanie potraktowany jako niepoprawny i zapisany w tabeli wynikowej,

- `message` – jeżeli `valid` jest `false`, to wówczas:
 - zawartość zmiennej zostanie zaprezentowana podczas wykonywania kroku oraz zapisana w tabeli wyjściowej w kolumnie `comment`.

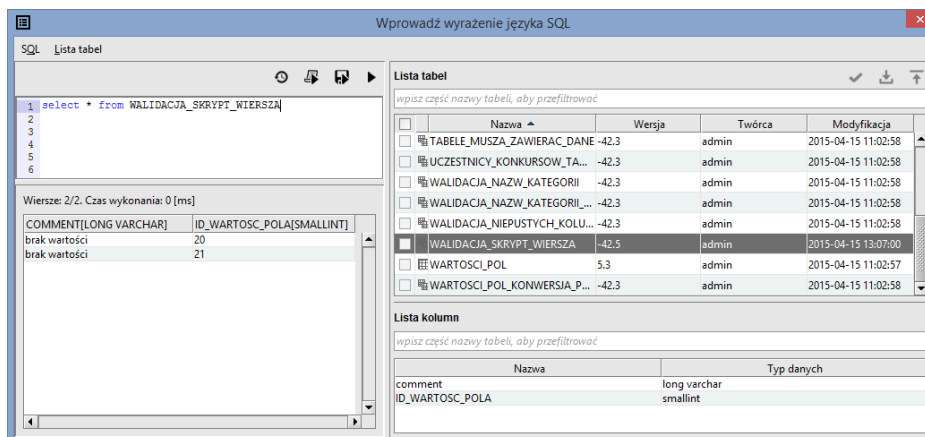
Panel **skrypt Groovy** (zob. Rysunek 121) zawiera:

- kod walidacji.

Sprawdzenie czy zawartość komórki jest pustym łańcuchem znaków zrealizowane zostało za pomocą metody `isEmpty()`.

```
if ( WARTOSCI_POL[ 'WARTOSC' ] == null || WARTOSCI_POL[ 'WARTOSC' ].isEmpty() )
{
    message = 'brak wartości'
    valid = false
}
```

Rysunek 122 przedstawia zawartość tabeli wyjściowej po wykonaniu zasilenia danymi.




The screenshot shows a window titled "Wprowadź wyrażenie języka SQL" with a sub-window "Lista tabel". The SQL query is `select * from WALIDACJA_SKRYPT_WIERSZA`. The execution result is displayed in a table with two columns: `COMMENT[LONG VARCHAR]` and `ID_WARTOSC_POLA[SMALLINT]`. The first row contains `brak wartości` and `20`, and the second row contains `brak wartości` and `21`. The execution time is 0 [ms].

COMMENT[LONG VARCHAR]	ID_WARTOSC_POLA[SMALLINT]
brak wartości	20
brak wartości	21

Rysunek 122. Walidacja wiersza skryptem - wynik wykonania

4.2.5 Instrukcja SQL

Wybranie funkcji „Instrukcja SQL” poprzez:

- ikonę: ,
- a następnie:
- instrukcja SQL.

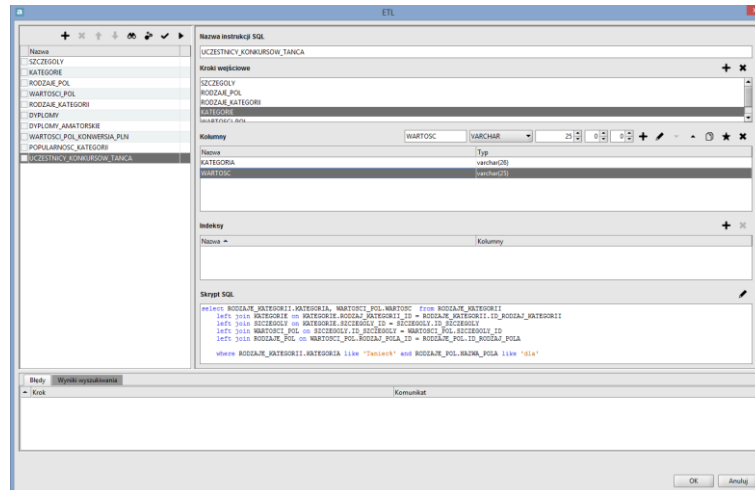
co spowoduje utworzenie nowego kroku, pozwalającego na:

- zdefiniowanie zapytania w języku SQL, które zostanie wykonane na tabelach wejściowych a wynik zostanie umieszczony w tabeli wyjściowej;

- wykonanie zapytania z wykorzystaniem wielu tabel wejściowych.

Wyniki są zapisywane w wyjściowej tabeli.

Przykładowe zapytanie typu „Instrukcja SQL” (zob. Rysunek 123):



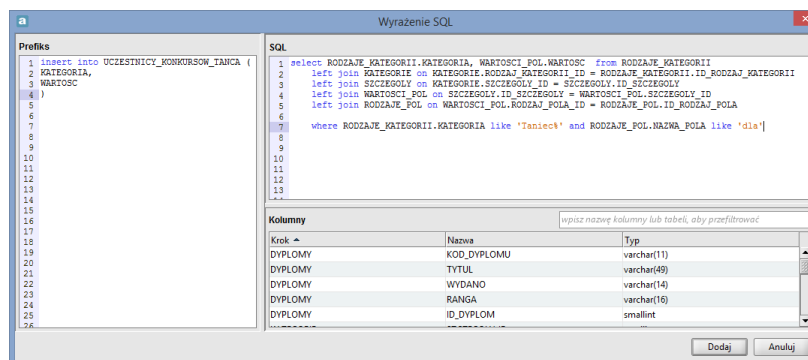
Rysunek 123. Instrukcja SQL- definicja kroku

Kolumny kroku muszą być zgodne z kolumnami wyspecyfikowanymi jako wynik operacji `select`.

Przykładowe zapytanie ma na celu uzyskanie listy osób, które uzyskały dyplom w kategoriach mających w nazwie słowo „taniec”.

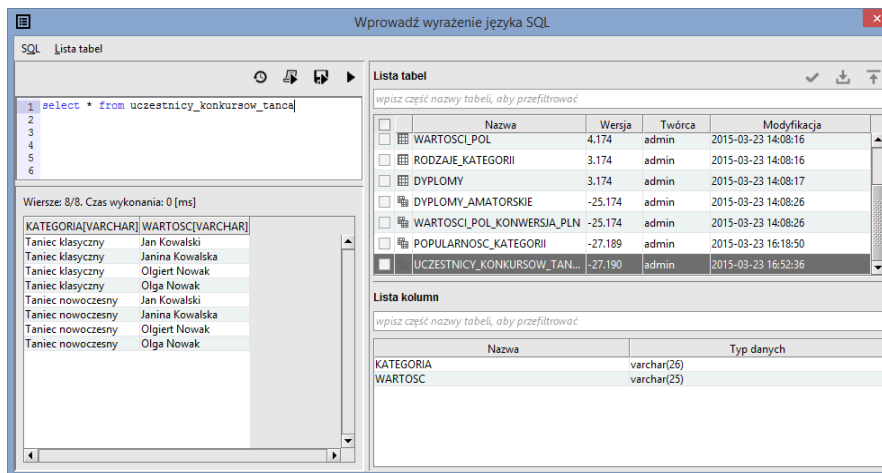
Instrukcja zapisu do bazy jest połączeniem:

- instrukcji `insert` przedstawionej w panelu **prefiks** oraz:
- `select` panelu **SQL** (zob. Rysunek 124).



Rysunek 124. Instrukcja SQL – panel edycji

Wynik zasilenia przedstawia Rysunek 125:




Rysunek 125. Instrukcja SQL - wynik wykonania

4.2.6 Walidacja tabeli

Poprawność danych zebranych w pojedynczej tabeli można sprawdzić za pomocą:

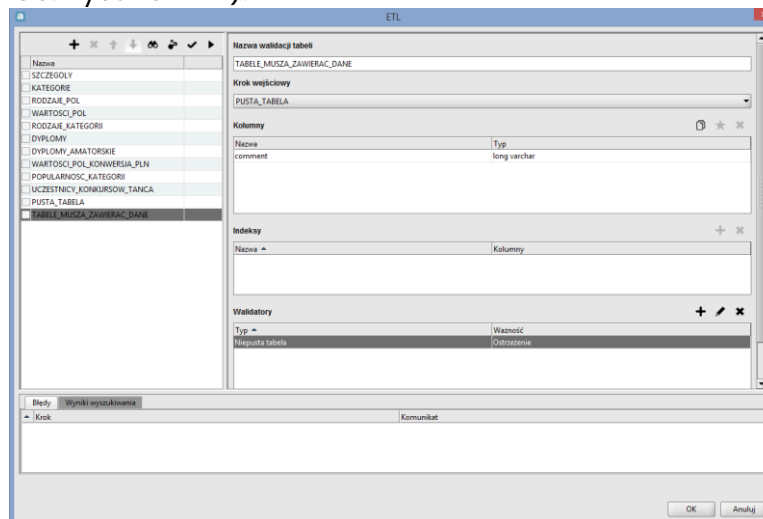
- kroku walidacja tabeli.

Wybranie funkcji „Walidacja tabeli” poprzez:

- Ikonę:  a następnie „Walidacja tabeli”, powoduje:

- utworzenie nowego kroku, pozwalającego na walidację całej tabeli z danymi za pomocą skryptów SQL i Groovy;

prezentację w prawym panelu okna odpowiednich pól i funkcji pozwalających na zdefiniowanie stosownej walidacji (zob. Rysunek 126).



Rysunek 126. Walidacja, niepusta tabela - definicja kroku

Dostępne są trzy rodzaje walidatorów:

- niepusta tabela,
- skrypt groovy,
- SQL.

Definiując którykolwiek z nich możliwe jest wybranie jego ważności.

Dla wartości ostrzeżenie proces przetwarzania danych nie zostaje przerwany (kolejny krok zostanie wykonany).

Krok walidacji, poza domyślną kolumną tabeli wyjściowej o nazwie `comment`, pozwala zdefiniować dodatkowe kolumny.

W pojedynczym kroku można definiować kilka walidatorów.

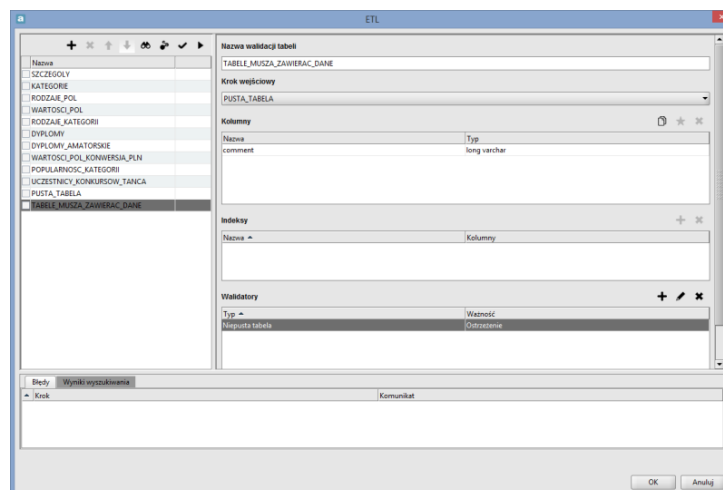
4.2.6.1 Niepusta tabela

Najprostszym sposobem na sprawdzenie tabeli jest zweryfikowanie czy krok wejściowy zawiera dane.

W przypadku, gdyby krok wejściowy zawierał dane, wówczas:

- tabela wyjściowa nie zawierałaby komentarzy (byłaby pusta).

Rysunek 127 przedstawia panel definicji kroku.



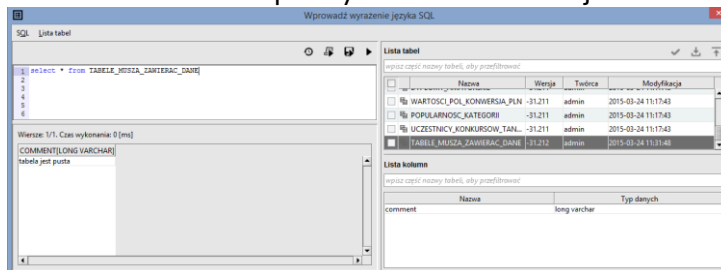
Rysunek 127. Walidacja, niepusta tabela - definicja kroku

Rysunek 128 przedstawia raport wykonania.



Rysunek 128. Walidacja, niepusta tabela - raport wykonania

Rysunek 129 przedstawia zawartość tabeli po wykonaniu walidacji.



Rysunek 129. Walidacja, niepusta tabela - wynik wykonania

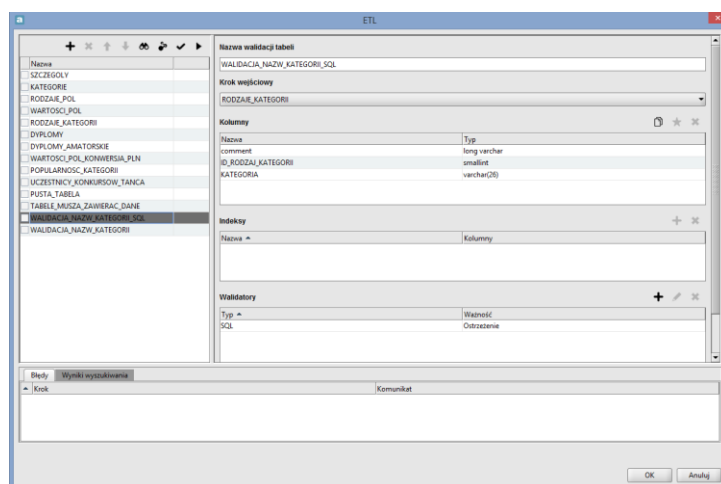
4.2.6.2 Zapytanie SQL

Walidacja tabeli za pomocą SQL polega na:

- zdefiniowaniu zapytania, które w przypadku poprawnych danych zwróci pusty wynik.

Jeżeli zapytanie zapisze rekordy do tabeli wyjściowej, zgłoszony zostanie błąd bądź ostrzeżenie.

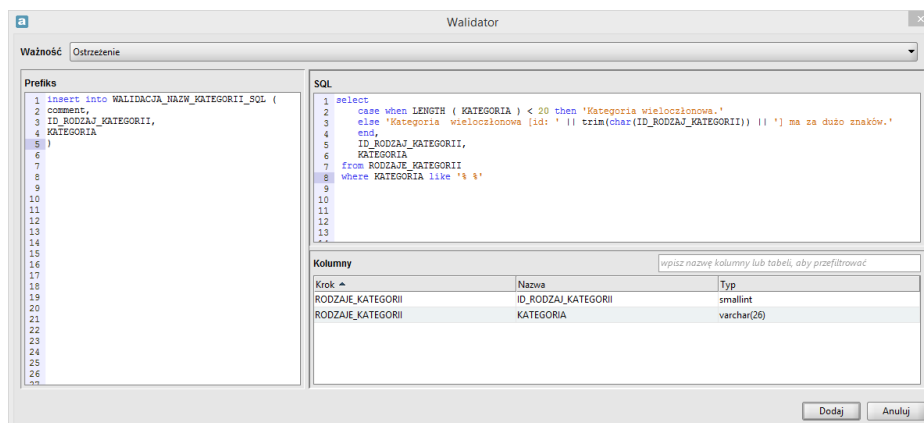
Panel ze zdefiniowanym krokiem walidacji z walidatorem typu SQL (zob. Rysunek 130):



Rysunek 130. Walidacja SQL - definicja kroku

Wynikowa tabela może zawierać kolumny będące kopiami kolumn kroku wejściowego.

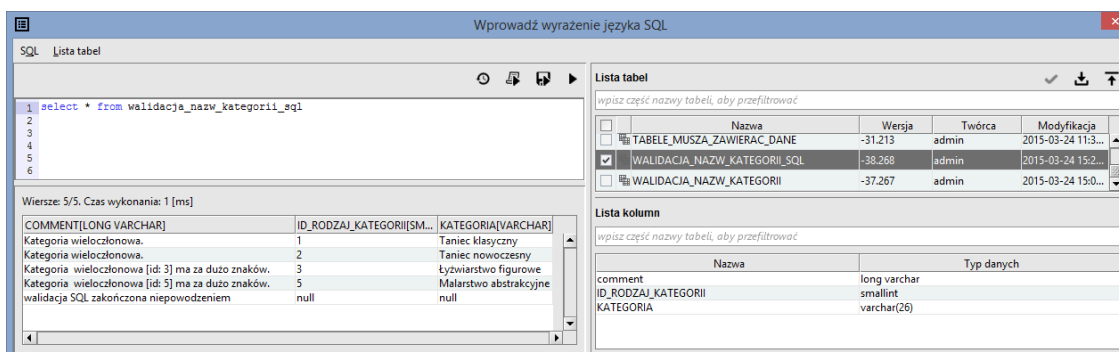
Zapytanie powstaje poprzez złożenie **prefiksu insert into** oraz **SQL – select** (zob. Rysunek 131).



Rysunek 131. Walidacja SQL - panel edycji

Tabelę uznaje się za prawidłową, jeżeli nazwy w kolumnie **KATEGORIA** nie zawierają spacji. Dodatkowo, inaczej raportowane są „niepoprawne” nazwy, jeżeli ich długość wynosi dwadzieścia lub więcej.

Należy zauważyć, że tabela wynikowa (zob. Rysunek 132) zawiera dodatkowy wiersz z komunikatem, niebędącym wynikiem definiowanego zapytania.



Rysunek 132. Walidacja SQL - wynik wykonania

4.2.6.3 Skrypt Groovy

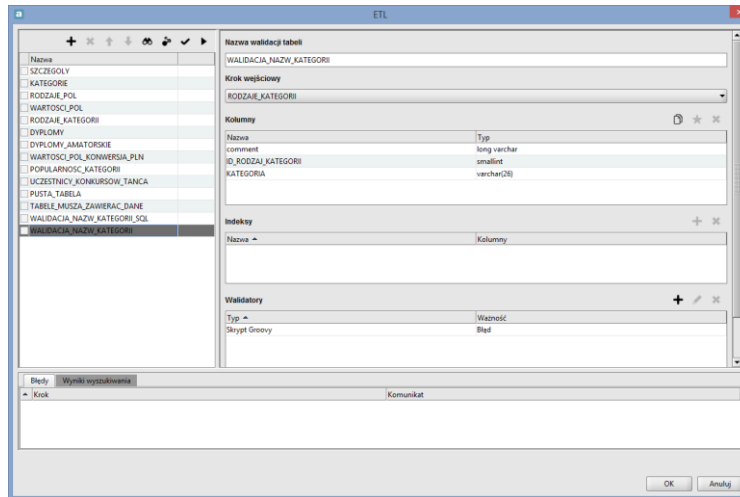
Krok „walidacja tabeli” umożliwia stworzenie:

- walidatora będącego skryptem **Goovy**.

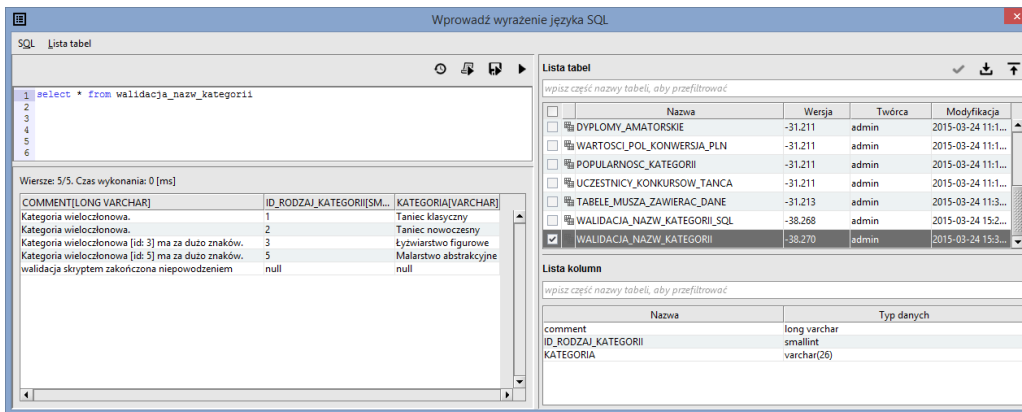
„Definicja kroku” (zob. Rysunek 133) oraz „wynik wykonania” (zob. Rysunek 134) są analogiczne jak w przykładzie z podrozdziału [Zapytanie SQL](#).

Skrypt realizuje to samo zadanie. Tabelę uznaje się za prawidłową, jeżeli nazwy w kolumnie **KATEGORIA** nie zawierają spacji.

Inaczej jednak raportowane są „niepoprawne” nazwy, jeżeli ich długość wynosi dwadzieścia lub więcej znaków.

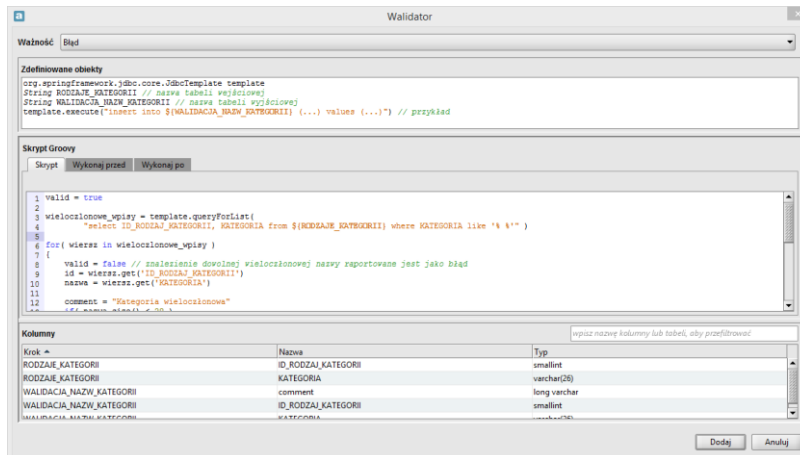


Rysunek 133. Walidacja Groovy – definicja kroku



Rysunek 134. Walidacja Groovy – wynik wykonania

Inaczej niż w przykładzie przedstawionym w podrozdziale [Zapytanie SQL](#), prezentowana jest zawartość panelu edycji przy walidacji Groovy (zob. Rysunek 135).



Rysunek 135. Walidacja Groovy – panel edycji

Okno do panelu edycji zostało podzielone na trzy części:

- **Zdefiniowane obiekty**
 - gdzie prezentowane są nazwy dostępne do tabel,
- **Skrypt Groovy,**
 - składający się z zakładek:
 - *Skrypt,*
 - *Wykonaj przed,*
 - *Wykonaj po,*
- **Kolumny**
 - prezentuje wszystkie kolumny jakie występują w tabelach.

Zapytania do bazy danych są realizowane poprzez obiekt `template` będący instancją klasy `JdbcTemplate`.

Nazwy tabel (oraz wartości zmiennych) są przekazywane do zapytań jako parametry, np. `${WALIDACJA_NAZW_KATEGORII}`. Dozwolonymi operacjami na tabeli wejściowej są zapytania `select`, natomiast na tabeli wynikowej – `insert into`.

Na potrzeby walidacji implementowana jest część **Skrypt**. Procedury **wykonaj przed** oraz **wykonaj po** nie są uruchamiane.

Przykład realizuje to samo sprawdzenie co w poprzednim rozdziale pojedyncze złożenie `insert into` oraz `select`. Skrypt rozdziela zapytanie, ponieważ walidator nie używa predefiniowanych komunikatów ani nie wykrywa błędów. Oznacza to, że wystąpienie danych w tabeli wynikowej nie generuje komunikatu o błędzie i nie przerywa sekwencji wykonywania kroków **ETL**. Odpowiednie zachowanie należy zaimplementować wraz z walidacją.

Na początku skryptu zadeklarowana i zainicjalizowana została zmienna `valid`. Na tym etapie, ma to znaczenie formalne; język **Groovy** nie pozwala na używanie nie zadeklarowanych zmiennych.

Odczyt niepoprawnych rekordów został zrealizowany za pomocą metody `queryForList()` z wykorzystaniem zapytania, które wybiera niepoprawne rekordy.

```
valid = true
wielozlonowe_wpisy = template.queryForList(
    "select ID_RODZAJ_KATEGORII, KATEGORIA from ${RODZAJE_KATEGORII} where KATEGORIA
like '% %'" )
```

Następnie, wiersze otrzymane jako wynik zapytania, zostają zanalizowane, w celu określenia sposobu raportowania błędów (w przykładzie ważna jest długość nazwy kategorii).

Po przygotowaniu docelowych wartości, następuje zapis do tabeli wynikowej.

```
for( wiersz in wieloczlonowe_wpisy )
{
    valid = false // znalezienie dowolnej wieloczlonowej nazwy raportowane jest jako bład
    id = wiersz.get('ID_RODZAJ_KATEGORII')
    nazwa = wiersz.get('KATEGORIA')

    comment = "Kategoria wieloczlonowa"
    if( nazwa.size() < 20 )
    {
        comment = comment + "."
    }
    else
    {
        comment = comment + " [id: " + id + "] ma za duzo znaków."
    }


    template.execute(
        "insert into ${WALIDACJA_NAZW_KATEGORII} (comment, ID_RODZAJ_KATEGORII, KATEGORIA)
values ('${comment}', ${id}, '${nazwa}')" )
}
```

W powyższym listingu została użyta zmienna specjalna `valid`. Została ona zainicjalizowana na początku skryptu wartością `true`. Jeżeli zmienna ta po wykonaniu skryptu będzie ustawiona na `false`, krok zostanie zakończony ze statusem błędu.

W ostatniej części skryptu zmienna `valid` zostaje sprawdzona. Jeżeli walidacja zakończy się niepowodzeniem, przygotowany zostaje komunikat. Przypisany zostaje on zmiennej specjalnej `message`. Zawartość zmiennej `message` zostaje zaprezentowana w podsumowaniu wykonania kroku oraz jest dopisywana do tabeli wynikowej w kolumnie `comment`. Jeżeli `valid` jest `true`, `message` nie jest wykorzystywany.

4.3 Usuwanie kroku

Aby usunąć błędny lub niepotrzebny krok, należy:



- zaznaczyć kwadrat występujący przed nazwą usuwanego kroku (przykład `TABLE_SCRIPT_0`),
- poprzez ikonę funkcyjną  dostępną nad listą tabel (kroków), wybrać funkcję „Usuń krok”.

Funkcja ta:

- usuwa bezpowrotnie zaznaczony krok,
- dotyczy tylko kroków (tabel) utworzonych w wyniku wykonanych przekształceń,
- nie jest aktywna dla tabel zdefiniowanych w wybranej migawce.

4.4 Zmiana kolejności prezentowania zdefiniowanych kroków

Aby zmienić kolejność prezentowania na liście zdefiniowanych kroków, należy:

- zaznaczyć kwadrat występujący przed nazwą kroku, który chcemy przesunąć (przykładowo: TABLE_SCRIPT_0);
- poprzez jedną z ikon funkcyjnych  , dostępnych nad listą tabel (kroków), wybrać funkcję „Zmień kolejność wykonywania kroku na wcześniejszą / późniejszą”.

Funkcje te:

- poprzez zmianę kolejności prezentowania tabel na liście, umożliwiają ustalenie wymaganej kolejności wykonywania poszczególnych kroków;
- dotyczą wszystkich tabel prezentowanych na liście.

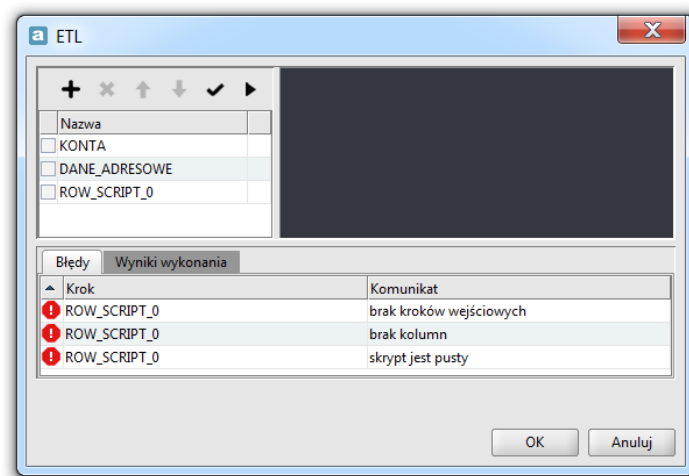
4.5 Walidacja zdefiniowanych kroków

Aby wykonać walidację zdefiniowanych kroków, należy wykorzystać funkcję „Waliduj kroki”, wybierając:

- ikonę: dostępną nad listą tabel (kroków).

Funkcja ta:


- może być uruchamiana w dowolnym momencie tworzenia kroków,
- wyniki walidacji prezentuje w dolnej części ekranu (zob. Rysunek 136).



Rysunek 136. ETL – wyniki walidacji kroków

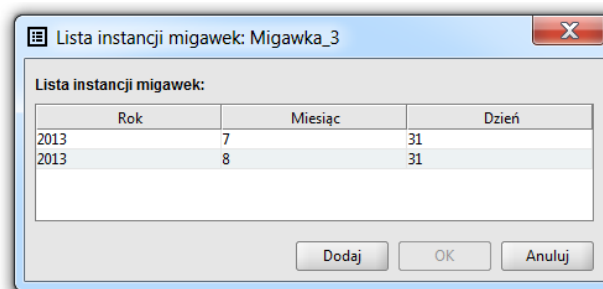
4.6 Testowe wykonanie zdefiniowanych kroków

Aby przeprowadzić test poprawności wykonania zdefiniowanych kroków, należy wykorzystać funkcję „Testowe wykonanie kroków”, poprzez wybranie:

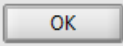
- Ikony:  , dostępnej nad listą tabel (kroków).

Funkcja ta:

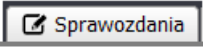
- może być uruchamiana w dowolnym momencie tworzenia kroków,
- poprzez przejście do „Zarządzania instancjami migawek” pozwala na testowe wykonanie zdefiniowanych kroków,
- prezentuje ekran jak po uruchomieniu funkcji „Zarządzaj instancjami migawek” (zob. Rysunek 137 i por. Rysunek 94).



Rysunek 137. ETL – lista instancji migawek dostępna przy testowym wykonywaniu zdefiniowanych kroków

Po wybraniu (podświetleniu) wymaganej instancji migawek, dla której mają zostać wykonane testy i zatwierdzeniu dokonanego wyboru klawiszem  uaktywniają się wszystkie funkcje dostępne z poziomu „Zarządzania instancjami migawek” (zob. rozdział [Zarządzaj instancjami migawek](#)).

5 Zasilanie według mapowań

Aby uruchomić funkcję „Zasilaj według mapowań”, należy wybrać zakładkę  , a następnie:

- Menu Sprawozdanie** → [Zasilaj wg mapowań](#)

Funkcja ta:

- jest dostępna tylko dla banków, posiadających licencję rozszerzoną o moduł „Mapowanie danych” i „Magazyn migawek”;
- pozwała na zasilanie aplikacji aSIS danymi, w oparciu o wyrażenia mapujące określone dla poszczególnych komórek sprawozdawczych i źródeł ich pobrania;
- dotyczy zawsze tylko jednego – aktywnego sprawozdania,

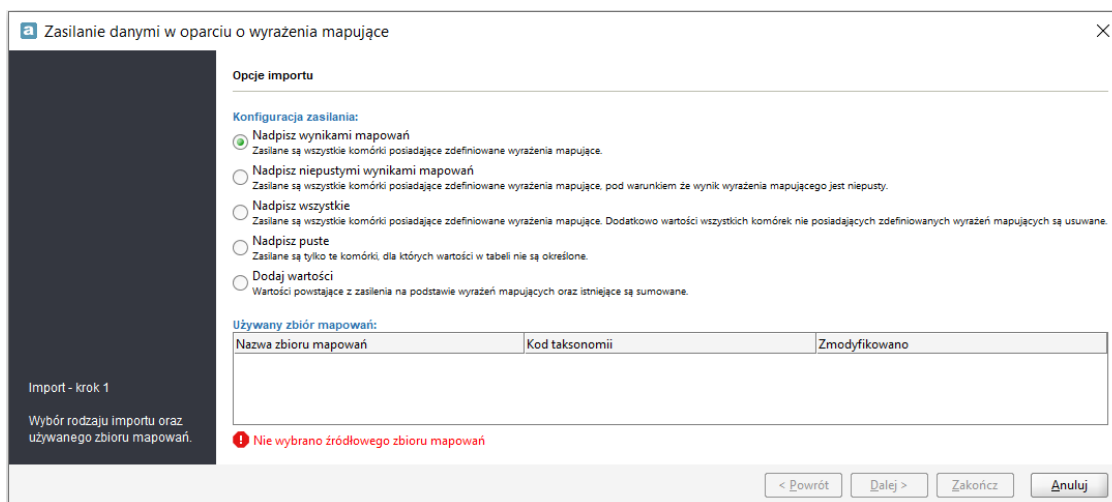
- może być wykonywana jednocześnie tylko na jednym stanowisku,
- jest aktywna tylko wtedy, gdy żaden z użytkowników nie pracuje przy module „Mapowanie danych” ani „Magazyn migawek”.

Po wybraniu funkcji „Zasilanie wg mapowań” wyświetla się ekran (zob. Rysunek 138).

Uruchomiona funkcja „Zasilania danymi wg mapowań”, ma podobny przebieg jak funkcja „Importuj dane” opisany w instrukcji [aSIST-Podstawy obsługi aplikacji.pdf](#) rozdział „Importuj dane”. Jednakże tutaj, wymagane jest od użytkownika wcześniejsze szczegółowe zdefiniowanie wyrażeń mapujących i źródeł zasilania danymi.

Pełny proces „Zasilania danymi wg mapowań” dokonywany jest w sześciu krokach.

Krok 1 – wybór opcji zasilania wg mapowań i zbioru mapowań (zob. Rysunek 138)



Rysunek 138. Opcje zasilania danymi po wybraniu funkcji "Zasilaj wg mapowań"

Możliwe jest tu wybranie jednej z opcji:

Nadpisz wynikami mapowań – co spowoduje:

- zasilenie wynikami mapowań wszystkich komórek posiadających zdefiniowane wyrażenia mapujące,
- zachowanie tylko tych wcześniej istniejących danych, które nie posiadają zdefiniowanych wyrażeń mapujących.

Nadpisz niepustymi wynikami mapowań – co spowoduje:

- zasilenie wynikami mapowań wszystkich tych komórek posiadających zdefiniowane wyrażenia mapujące, dla których wynik mapowań jest niepusty;
- zachowanie tych wcześniej istniejących danych, które nie posiadały zdefiniowanych wyrażeń mapujących lub wynik mapowań był pusty.

Nadpisz wszystkie – co spowoduje:

- zasilenie wynikami mapowań wszystkich komórek posiadających zdefiniowane wyrażenia mapujące,
- usunięcie wartości tych komórek, które nie posiadają zdefiniowanych wyrażeń mapujących.

Nadpisz puste – co spowoduje:

- zasilenie wynikami mapowań tylko tych komórek, które nie miały wprowadzonych żadnych wartości;
- zachowanie wszystkich danych istniejących przed wykonaniem opcji „zasilaj według mapowań”.

Dodaj wartości – co spowoduje:

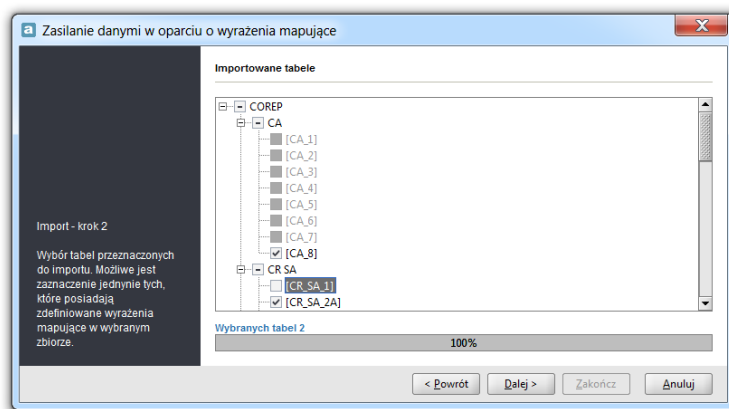
- zsumowanie wartości powstających z zasilenia na podstawie wyrażeń mapujących (o ile nie są puste) oraz wartości istniejących;
- wartości powstające z zasilenia i wcześniej istniejące dane będą sumowane.

W dolnej części ekranu, wyświetla się lista wszystkich dostępnych zbiorów z wyrażeniami mapującymi i konieczne jest wskazanie jednego z nich.

Na liście tej widnieją tylko:

- zbiory dotyczące sprawozdania aktualnie zasilanego,
- zbiory o statusie „aktywne” (deaktywowane zbiory do mapowań są tu niewidoczne).

Jeśli lista jest pusta, to znaczy, że dla aktywnego sprawozdania aplikacja nie widzi żadnego zbioru z wyrażeniami mapującymi.

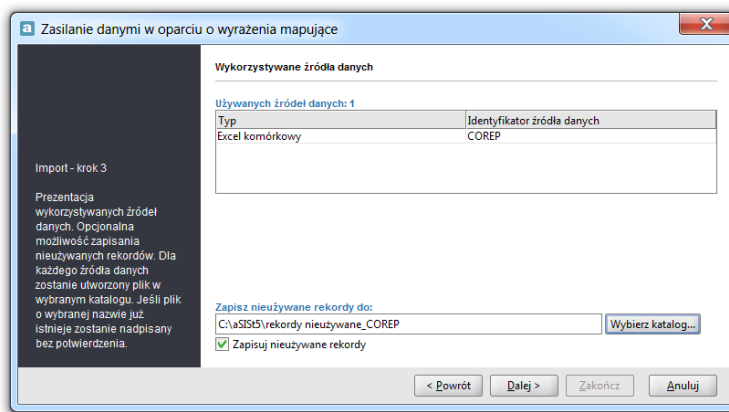
Krok 2 – wybór tabel do zasilenia wg mapowań (zob. Rysunek 139)

Rysunek 139. Wybór tabel do "Zasilania wg mapowań"

W formie drzewa prezentowane są tutaj tabele, które mogą zostać zasilone danymi.

- oznacza, że dana tabela zostanie zasilona danymi (Rysunek 139: CA_8, CR_SA_2A),
- oznacza, że zrezygnowaliśmy z zasilenia danymi tabeli, znajdującej się obok (Rysunek 139: CR_SA_1);
- oznacza, że dla tej tabeli w zbiorze wskazanym w kroku 1, nie ma zdefiniowanych żadnych wyrażeń mapujących i tabela ta nie zostanie zasilona danymi wg wyrażeń.

Krok 3 – prezentacja wykorzystywanych źródeł danych (zob. Rysunek 140 i Rysunek 141)



Rysunek 140. Prezentacja wykorzystywanych źródeł danych i wybór opcji „śledzenia nieużywanych rekordów”

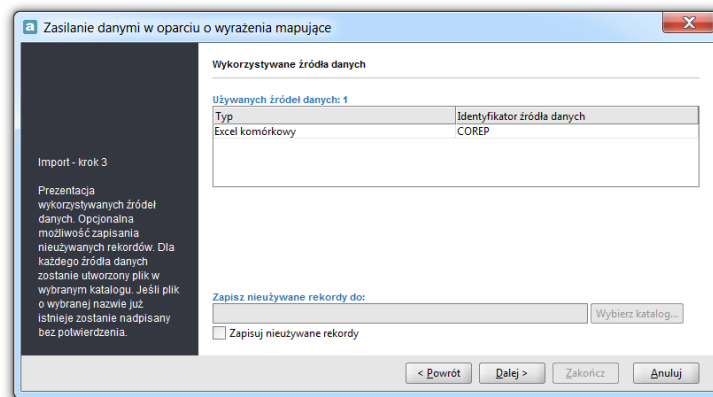
W kroku tym, prezentowana jest informacja o wykorzystywanych źródłach danych.

Dodatkowo istnieje tu możliwość zaznaczenia opcji, pozwalającej w kroku szóstym, zaprezentować ewentualną listę nieużywanych rekordów.

Zaznaczenie pozycji **Zapisuj nieużywane rekordy** (zob. Rysunek 140), uaktywni przycisk **Wybierz katalog...** i pozwoli na wskazanie miejsca zapisu plików z niewykorzystanymi rekordami (dla każdego źródła danych – odrębny plik).

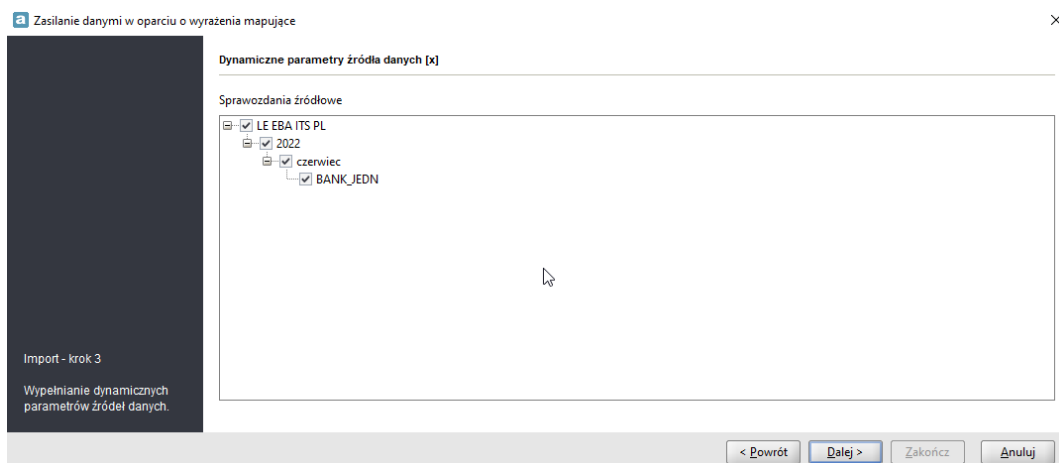
Jeżeli pozycja **Zapisuj nieużywane rekordy** nie zostanie zaznaczona (zob. Rysunek 141) wówczas:

- nie będzie wykorzystana opcja „śledzenia nieużywanych rekordów”,
- nie będzie informacji o tym, czy wszystkie rekordy zostały wykorzystane podczas zasilania danymi.



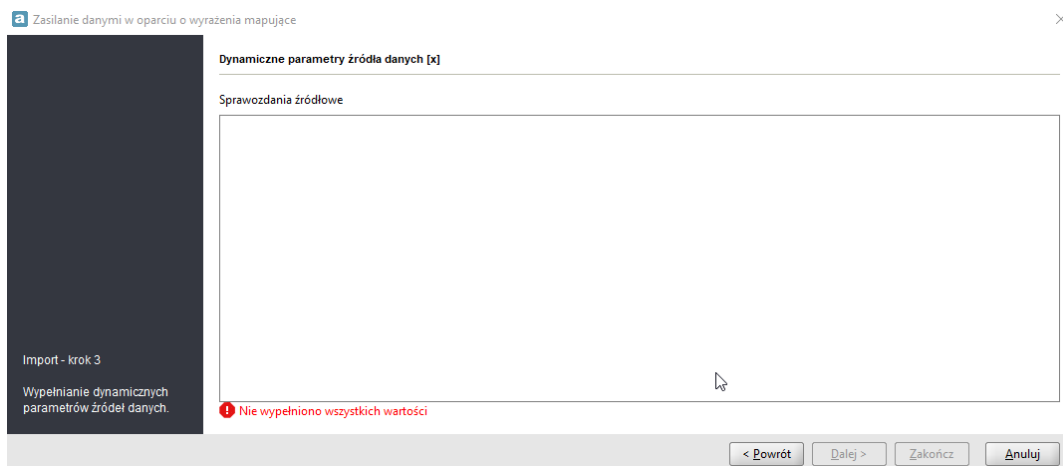
Rysunek 141. Prezentacja wykorzystywanych źródeł danych

W przypadku sprawozdania z okresu względnego, kreator zasilania danymi wskaże dane na podstawie pola „Okres względny” dostępnego przy tworzeniu i edycji źródła danych (zob. rozdział [Wewnętrzne źródło danych – typy sprawozdań źródłowych.](#)).



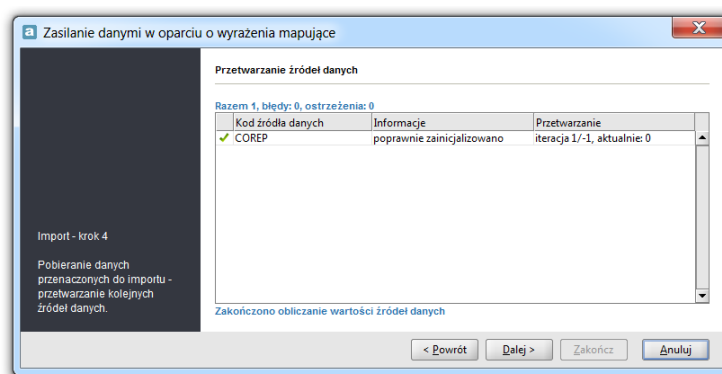
Rysunek 142. Wypełnianie dynamicznych parametrów źródeł danych

W przypadku braku sprawozdania za wskazany okres (np. dla wartości **1M** będzie to poprzedni miesiąc) pojawi się komunikat:



Rysunek 143. Wypełnianie dynamicznych parametrów źródeł danych – brak danych

Krok 4 – pobieranie danych przeznaczonych do zasilenia wg mapowań (zob. Rysunek 144).

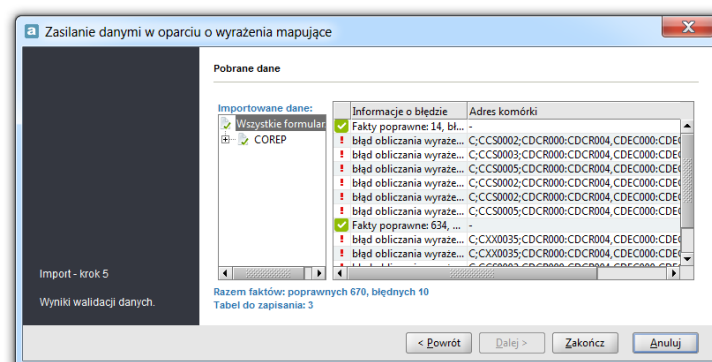


Rysunek 144. Zasilania wg mapowań – pobieranie danych do zasilenia

W prezentowanym oknie wyświetlona jest:

- informacja o błędach wykrytych podczas walidacji,
- kod źródła danych, w oparciu o który dokonywany jest import danych wg mapowań.

Krok 5 – prezentacja wyników dokonanej walidacji (zob. Rysunek 145).

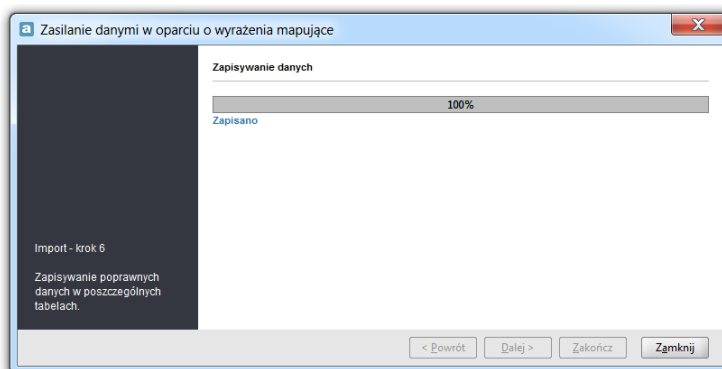


Rysunek 145. Prezentacja wyników walidacji

Szczegółowe wyniki dokonanej walidacji, odnoszą się tutaj do:

- wyświetlonej listy tabel, do których mogą zostać zaimportowane dane oraz
- informacja o błędach wykrytych podczas walidacji.

Krok 6 – zapisywanie poprawnych danych do poszczególnych tabel (zob. Rysunek 146).

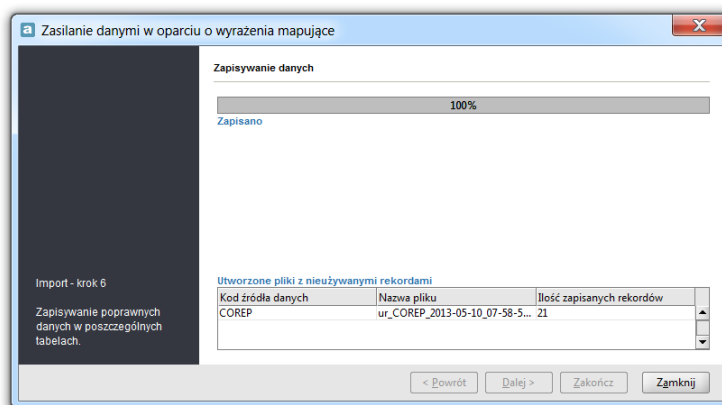


Rysunek 146. Zapisywanie poprawnych danych do tabel

Następuje tu zapisywanie poprawnych danych do tabel ze zdefiniowanymi wyrażeniami mapującymi. Błędne dane są zawsze pomijane.

Jeśli w kroku 3, zaznaczona była opcja śledzenia nieużywanych rekordów, wówczas dodatkowo w dolnej części ekranu w formie tabeli (zob. Rysunek 147), wyświetlana jest informacja o utworzonych plikach z takimi niewykorzystanymi rekordami, z wyszczególnieniem:

- kodu źródła danych,
- nazwy utworzonego pliku,
- ilości nieużywanych rekordów, zapisanych do tego pliku.



Rysunek 147. Zapisywanie poprawnych danych i nieużywanych rekordów

Tabela ta jest pusta, gdy wszystkie dane były poprawne.

6 Informacje dodatkowe

6.1 Źródła danych

Źródła danych to źródła, do których odwołują się zdefiniowane wyrażenia mapujące dane.

Definiowane są globalnie dla całej aplikacji i są współdzielone przez wykorzystujące je wyrażenia.

Posiadają unikatowy identyfikator, nazywany dalej kodem źródła danych.

Tabelaryczne źródło danych – opisuje dane przedstawione w postaci poprawnej tabeli, gdzie:

- pierwszy wiersz reprezentuje nagłówki tabel, a wszystkie pozostałe – dane;
- każda kolumna ma unikatowy nagłówek (który jest poprawnym identyfikatorem zgodnym z formatem:
`[a-zA-Z][a-zA-Z0-9_]*`);
- wszystkie kolumny mają taką samą liczbę wierszy.

6.1.1 Tabelaryczne źródła danych typu Excel

Poprawnie zdefiniowane źródło typu Excel posiada:

- ścieżkę do pliku zgodnego z formatem Excel 2003,
- numer arkusza (arkusze numerowane od 1),
- poprawne dane tabelaryczne rozpoczynające się w pierwszej kolumnie i pierwszym wierszu wskazanego arkusza;
- może posiadać puste wiersze – będą one pomijane w czasie wczytywania pliku.

6.1.2 Tabelaryczne źródła danych typu CSV

Poprawnie zdefiniowane źródło posiada:

- ścieżkę do pliku zgodnego z formatem CSV,
- nazwy kolumn w pierwszym wierszu pliku,
- pola oddzielone jednym z wybieralnych separatorów.

6.2 Wyrażenia mapujące

Wyrażenia mapujące to formuły odwołujące się do zdefiniowanych źródeł danych, w celu pobrania z nich wartości i opcjonalnie wykonania działań dodatkowych (takich jak np. suma dla wartości liczbowych).

Z wyjątkiem omówionych poniżej sytuacji szczególnych, pojedyncze wyrażenie może korzystać z dowolnej liczby źródeł danych.

Źródła danych muszą być uprzednio zdefiniowane, aby można było wskazać je w wyrażeniu mapującym.

6.2.1 Typy komórek sprawozdawczych

Przy tworzeniu wyrażen mapujących, dozwolone są cztery typy formułowanych wyrażen. Typy te odpowiadają typom mapowanych komórek:

- LICZBOWY:

Wyrażenie liczbowe jest złożeniem dowolnych działań arytmetycznych.

Dozwolone jest dzielenie i mnożenie zmiennoprzecinkowe, dodawanie, odejmowanie oraz zagnieżdżanie w nawiasach.

Wyrażenie liczbowe może odwoływać się do dowolnej ilości wartości wyliczanych w źródłach danych.

Można również używać stałych liczbowych z kropką jako separatorem dziesiętnym.

<wyrażenie arytmetyczne> ::= (<argument arytmetyczny> | <operacja arytmetyczna>)

<argument arytmetyczny> ::= (<liczba> | <wartość źródła danych>)

<operacja arytmetyczna> ::= (<operacja arytmetyczna><operator arytmetyczny><argument arytmetyczny> | "("<operacja arytmetyczna> ")" | "-" <operacja arytmetyczna>) | "+" <operacja arytmetyczna>)

<operator arytmetyczny> ::= ("/" | "*" | "+" | "-")

- LOGICZNY:

Wyrażenie logiczne - to dowolne wyrażenie używające logicznych wartości źródeł danych, operatorów logicznych oraz stałych logicznych (**true**, **false**).

Operatory logiczne (<&&> i <||>) oznaczają odpowiednio koniunkcję i alternatywę logiczną, gdzie koniunkcja ma wyższy priorytet.

Możliwe jest również dowolne zagnieżdżanie w nawiasach. Wyrażenia w nawiasach posiadają najwyższy priorytet.

<wyrażenie logiczne> ::= (<argument logiczny> | <operacja logiczna>)

<argument logiczny> ::= (<stała logiczna> | <wartość źródła danych>)

<operacja logiczna> ::= (<operacja logiczna><operator logiczny><argument logiczny> | "("<operacja logiczna > ")")

<operator logiczny > ::= ("&&" | "||")

<stała logiczna> literał **true** i **false** (odpowiednio oznaczający prawdę i fałsz), dopuszczalne są dowolne wielkości liter

- TEKSTOWY:

Wyrażenie tekstowe – to stała tekstowa lub tekstowa wartość źródła danych.

<wyrażenie tekstowe> ::= (<stała tekstowa> | <wartość źródła danych>)

<stała tekstowa> dowolny ciąg znaków objęty w pojedynczy cudzysłów.

Znakiem ucieczki w literałach tekstowych jest "\", a więc aby w treść napisu wstawić "\"" to należy poprzedzić go znakiem ucieczki.

Wszystkie inne znaki poprzedzane tym znakiem będą wstawiane bez zmian.

Jednocześnie zawartość poprawnego napisu nie może być zakończona żadnym z obu znaków specjalnych.

Cudzysłów może zostać pominięty w napisach zaczynających się od litery, nieposiadających znaków białych i nie będących słowami kluczowymi.

- DATA:

Wyrażeniem typu data może być jedynie wartość jednego źródła danych o takim samym typie.

<wyrażenie data> ::= <wartość źródła danych>

Typ wyrażenia wybierany jest automatycznie (zgodnie z typem znajdującym się w taksonomii) i staje się podstawą kontroli typu edytowanego wyrażenia.

Wszystkie występujące w nim źródła danych również zobligowane są do zwracania wartości zgodnej z określonym dla całego wyrażenia typie.

Wszystkie z powyższych wyrażen korzystają z opisu wartości pobieranej ze źródła zgodnie z poniższą składnią:

<identyfikator_źródła>:{ <wyrażenie> }

<identyfikator_źródła> unikatowy identyfikator źródła danych (podawany przy tworzeniu) – musi mieć format poprawnego identyfikatora opisanego wyrażeniem regularnym [a-zA-Z][a-zA-Z0-9_]*

<wyrażenie> opisane w dalszej części rozdziału, obliczające wartość zwracaną przez wskazane źródło danych.

Na przykład:

ds:{1}

gdzie 'ds' to identyfikator a '1' to wyrażenie.

Typ wyrażenia wybierany jest automatycznie (zgodnie z typem znajdującym się w taksonomii) i staje się podstawą kontroli typu edytowanego wyrażenia.

Najczęściej wyrażenia mapujące dane sprawozdawcze zawierają wartości liczbowe.

W takich przypadkach wyrażenie mapujące wykorzystuje wartości liczbowe i dostępne dla nich operatory:

- <+> - oznaczające dodawanie,
- <-> - oznaczające odejmowanie,
- <*> - oznaczające mnożenie,
- </> - oznaczające dzielenie

Operatory te są łączone zgodnie z przyjętymi zasadami – mnożenie i dzielenie przed dodawaniem i odejmowaniem.

W przypadku bardziej złożonych obliczeń w celu wymuszenia prawidłowego łączenia argumentów działania można używać operatorów nawiasów <(> i < >). Dlatego zapis:

$2 * (2 + 2)$ w wyliczeniu da wynik 8

Niektóre komórki sprawozdawcze posiadają jednak dane innych typów.

W aplikacji aSISt występują komórki, które w module mapowań traktowane są jako wartości tekstowe:

- edytowane z listy wybieranych wartości (np. waluty) lub
- wpisywane bezpośrednio (np. nazwa jednostki).

W przypadku takich komórek wartość można jedynie pobrać bezpośrednio ze źródła danych (operacje arytmetyczne nie są dozwolone) lub użyć stałej tekstowej.

Komórka sprawozdawcza może również być **typu daty** (w takim przypadku w aplikacji aSISt do edycji służy kalendarz) – wtedy do mapowania można użyć jedynie pojedynczego źródła danych, z którego pobrana będzie data.

6.2.2 Rodzaje wyrażień mapujących

Wyróżnia się trzy rodzaje wyrażień mapujących:

- **Wyrażenia mapujące pojedyncze komórki sprawozdawcze** (nie listowe) – taka komórka może posiadać maksymalnie jedno wyrażenie mapujące odwołujące się do dowolnej ilości zdefiniowanych źródeł danych.
- **Wyrażenie mapujące komórki tabel listowych lub wymiarów użytkownika.** Takie wyrażenie ma postać podobną do wyrażenia mapującego zwykłe komórki sprawozdawcze, jednak liczba wartości uzyskanych przez jego wyliczenie jest równa liczbie rekordów listy lub odpowiednio liczbie wartości wymiarów użytkownika uzyskanych przy procesie zasilania.
- **Wyrażenie grupujące.**
Wyrażenie wpisywane w specjalne pole dla tabel listowych lub przy definiowaniu wymiarów użytkownika, które w czasie zasilania wyznacza liczbę uzyskanych rekordów listy lub wartości wymiarów użytkownika. Takie wyrażenie musi odwoływać się do dokładnie jednego źródła danych i używać jednej z dostępnych funkcji grupujących.

6.2.2.1 Mapowanie pojedynczych komórek sprawozdawczych

W najprostszym przypadku:

- wyrażenie mapujące odwołuje się do jednego źródła danych i
- obliczona przez źródło danych wartość jest zapisywana w mapowanej komórce.

W takim przypadku wyrażenie mapujące ma postać:

def:{ <wyrażenie agregujące > }

gdzie:

def – jest to identyfikator źródła danych

wyrażenie agregujące – wybiera wymagane rekordy źródła danych i wykonuje na nim wymagane obliczenia.

Przykładowo:

Sumowanie sald kont w wymaganej walucie dla źródła danych o symbolu dane, posiadającego kolumny o nazwie saldo i waluta mogą wyglądać następująco:

dane:{ sum(\$saldo, \$waluta = PLN) }

W celu zasilenia komórki średnią sumą sald w PLN z dwóch różnych źródeł dane1 i dane2 wymaga mapowania:

(dane1:{ sum(\$saldo, \$waluta = PLN) } + dane2:{ sum(\$saldo, \$waluta = PLN) }) / 2

6.2.2.2 Pobieranie danych ze źródła

Źródła danych są tabelarycznymi zbiorami danych, gdzie każda kolumna posiada unikatową nazwę.

W wyrażeniach mapujących w celu odwołania do wartości danej kolumny należy użyć jej nazwy poprzedzonej znakiem <\$>

Przykładowo: `$saldo_winien`

Wyrażenia źródła danych (część wyrażenia znajdująca się pomiędzy <{ i <}>) stara się obliczyć pojedynczą wartość na podstawie określonej ilości rekordów ze źródła.

W tym celu wykorzystywane są funkcje agregujące. Każda z dostępnych funkcji agregujących wylicza jedną wartość.

Wszystkie funkcje agregujące można zapisać schematycznie jako

<nazwa_funkcji>(<wyrażenie>, <dyskryminator>)

<wyrażenie> to w ogólności dowolne wyrażenie będące podstawą obliczania funkcji agregującej. Najczęściej jest to pojedyncza kolumna źródła danych (np. `$saldo`).

Przykładowo, w celu poszukiwania największego salda można zapisać:

`max($saldo)`

Często jednak w ramach zainteresowania danej funkcji agregujących znajduje się tylko część rekordów zbioru. W tym celu można w funkcji agregującej podać dodatkowy warunek (**<dyskryminator>**).

Przykładowo:

`max($saldo, $waluta = PLN)`

będzie poszukiwać największego salda tylko w wierszach, które w kolumnie `$waluta` mają wartość PLN.

W bardziej złożonych przypadkach może okazać się, że pojedynczy warunek nie jest wystarczający. W przypadku, kiedy wymagane jest sumowanie kolumny `$saldo_ma` dla PLN i dla tych rekordów, które w kolumnie `$saldo_winien` mają wartość większą od zero. Wtedy należy zapisać

`sum($saldo_ma, $saldo_winien > 0 && $waluta = PLN)`

W tym przypadku użyto operatora logicznego, sprawdzającego oba warunki jednocześnie (koniunkcja). Mechanizmy konstruowania złożonych warunków przedstawiono w dalszej części rozdziału.

Dostępne **funkcje agregujące** to: `sum`, `avg`, `first`, `last`, `min`, `max`, `count`, `same`, `one`, `any` (zob. rozdział [Funkcje agregujące](#)).

6.2.2.3 Mapowanie komórek listowych

Wyrażenia mapujące **w tabelach listowych** mają postać podobną do powyżej omówionych, jednak istnieją pewne różnice związane ze sposobem ich interpretacji.

Natura tabel listowych sugeruje, że ilość elementów listy nie jest znana – zależy ona na przykład od ilości i rodzajów transakcji w danym okresie sprawozdawczym.

W związku z tym istnieje potrzeba za pomocą jednego wyrażenia mapującego wyliczenia wszystkich wartości (po jednej dla każdego uzyskiwanego w tabeli listowej rekordu).

Aby takie działanie było możliwe konieczne jest zdefiniowanie wyrażenia grupującego.

Wyrażenie grupujące w przypadku tabel listowych znajduje się w pierwszej kolumnie (pierwszym wierszu – w zależności od orientacji tabeli).

W sprawozdaniu taka komórka jest nieedytowalna.

Wyrażenie grupujące **musi wykorzystywać dokładnie jedno źródło danych** i ma postać:

def:{ funkcja_grupująca(<argumenty>) }

Funkcje grupujące **nie obliczają wartości, ale wyznaczają grupy wierszy**, które są podstawą wyznaczania wartości innych wyrażeń znajdujących się w grupie.

W przypadku tabel listowych grupę obejmują wszystkie wyrażenia w ramach jednego rekordu listy.

Istnieją dwie funkcje grupujące:

- **group** – grupowanie, przyjmuje niepustą listę wyrażeń grupujących dowolnego typu.

Każdy wektor wartości argumentów wyznacza grupę,

- **conditionGroup** – grupowanie warunkowe.

Jako pierwszy argument przyjmuje warunek, który spełniają wszystkie grupy wyznaczone przez funkcję. Kolejne argumenty to lista wyrażeń grupujących dowolnego typu.

Każdy wektor wartości argumentów wyznacza grupę.

6.2.2.4 Mapowanie wymiarów użytkownika

W przypadku mapowania wymiarów użytkownika, wymagane jest wyznaczenie wartości wymiarów użytkownika, które zostaną obliczone w procesie zasilania.

Podobnie jak przy mapowaniu komórek listowych, tutaj również **konieczne jest zdefiniowanie wyrażeń grupujących, obliczających tyle wartości komórek wynikowych, ile zostało obliczonych różnych wartości wymiaru użytkownika.**

Do tego celu wykorzystywane są również dwie funkcje grupujące:

- **group** – grupowanie wyrażeń zdefiniowanego typu,
- **conditionGroup** – grupowanie warunkowe.

6.2.3 Kontrola poprawności wyrażeń mapujących

Budowane wyrażenia mapujące podlegają każdorazowo kontroli, czy:

- są zgodne z przyjętą składnią,
- są poprawne ze względu na typ komórki sprawozdawczej,
- wszystkie użyte identyfikatory źródeł danych są poprawne (istnieją takie źródła danych).

W przypadku wykrycia przez system nieprawidłowości, każdorazowo jesteśmy informowani o tym odpowiednim komunikatem o przyczynie błędu.

Jeżeli edycja zostanie zakończona poprawnie, dokonywane jest automatyczne formatowanie wprowadzonego wyrażenia, czyli:

- wyrównanie wprowadzonego tekstu,
- usunięcie nadmiarowych nawiasów,
- usunięcie wielokrotnego zagnieżdżenia operatora '-' lub '+' przed literałami liczbowymi,
- usunięcie drugiego argumentu funkcji agregującej, jeżeli jest literałem i ma wartość true;
- usunięcie formalnego zapisu funkcji same, jeśli nie posiada dodatkowych warunków i nie jest zagnieżdżona w żadnej innej funkcji agregującej;
- usunięcie nadmiarowego wstąpienia rzutowania to_number (w przypadku, kiedy typ wyrażenia jest jednoznaczny),
- usunięcie formatu daty z rzutowania to_date w przypadku, kiedy jest on domyślny ('yyyy-MM-dd').

6.2.4 Składnia języka wyrażeń grupujących

Wyrażenie grupujące wskazuje dokładnie jedno źródło danych zawierające wyrażenie grupujące:

```
<wyrażenie grupujące> ::= <kod źródła danych>:"{" <wyrażenie grupujące źródła danych> "}"
```

Wyrażenia grupujące nie obliczają wartości komórek, ale wyznaczają grupy wartości będące podstawą wartościowania innych wyrażeń tego źródła.

Wyrażenia zależne od wyrażenia grupującego generują osobny zbiór wartości dla każdej grupy wyznaczonej przez to wyrażenie.

Własność ta jest wykorzystywana przy mapowaniu wymiarów użytkownika i tupli w celu stworzenia wielu grup wartości dla jednej grupy wyrażeń.

6.2.5 Obliczanie wartości wyrażenia mapującego

Obliczanie wartości wyrażenia mapującego wiąże się z obliczeniem wszystkich wartości wyrażeń użytych źródeł danych.

Każde z użytych źródeł danych musi opisywać dokładnie jedną wartość.

W jednym wyrażeniu może znajdować się dowolna ilość wyrażeń korzystających z różnych źródeł danych, jak również dane źródło może być wykorzystane do dostarczenia wielu wartości.

Zgłoszenie błędu przez przynajmniej jedno używane źródło danych powoduje błąd obliczania całego wyrażenia.

W przypadku wyrażeń objętych wyrażeniem grupującym wszystkie podwyrażenia używające tego samego źródła danych co wyrażenie grupujące stanowią podstawę generowania wielu wartości dla jednego wyrażenia mapującego.

Wartościowanie takiego wyrażenia odbywa się wielokrotnie – dla każdej grupy wyznaczonej przez wyrażenie grupujące.

6.2.6 Język tabelarycznych źródeł danych

6.2.6.1 Gramatyka

Tabelaryczny język wyrażeń źródeł danych (Excel i CSV) wspiera dwa typy wyrażeń:

- **"wyrażenia tabelaryczne"** obliczające wartości i
- **"grupujące wyrażenie tabelaryczne"** wyznaczające grupy będące podstawą wartościowania innych wyrażeń tabelarycznych.

Język definiuje wyrażenia zgodne z gramatyką:

<wyrażenie tabelaryczne> ::= <operacja>

<operacja> ::= (<argument> | <operacja arytmetyczna> | <operacja porównania> | "(" <operacja> ")")

<operacja arytmetyczna> ::= (<operacja><operator arytmetyczny><argument> | <operacja><operator logiczny><argument>)

<operator arytmetyczny> ::= ("+" | "-" | "/" | "*")

<operator logiczny> ::= ("||" | "&&")

<operacja porównania> ::= (<operacja><operator porównania><operacja> | <operacja><operator porównania><zbiór> | <zbiór><operator porównania><operacja>)

<operator porównania> ::= ("=" | "!=" | "<" | "<=" | ">" | ">=")

<zbiór> ::= ("[" "]" | "[" <operacja> ["," <operacja>] * "]")

<argument> ::= (<kolumna> | <litera> | <funkcja> | "*")

<litera> ::= (<stała liczbowa> | <stała logiczna> | <stała tekstowa>)

<funkcja> ::= <nazwa funkcji> "(" [<argumenty funkcji>] ")"

<argumenty funkcji> ::= <operacja> ["," <operacja>] *

<stała liczbowa> dowolna stała liczbowa z "." jako separatorem dziesiętnym

<stała logiczna> **true** (prawda) lub **false** (fałsz) – litery dowolnej wielkości

<stała tekstowa> dowolny ciąg znaków alfanumerycznych bez znaków białych nierozpoczynający się od liczby lub dowolny ciąg znaków w pojedynczych cudzysłowach (""). Znakiem ucieczki jest "\". Jego stosowanie pozwala zastosować znaki specjalne jako zwykłe znaki w napisie. Stosowanie znaku ucieczki przed znakami niespecjalnymi jest ignorowane.

<kolumna> poprawny identyfikator kolumny poprzedzony znakiem "\$"

<nazwa funkcji> jedna z akceptowanych nazw funkcji operujących na wartościach, agregujących lub podwójnie agregujących.

<grupujące wyrażenie tabelaryczne> ::= <nazwa funkcji grupującej> "(" [<argumenty funkcji>] ")"

<nazwa funkcji grupującej> jedna z akceptowanych nazw funkcji grupujących

6.2.6.2 Operacje

Jak wcześniej wspomniano istnieje możliwość budowania złożonych warunków ograniczających wybierane rekordy w funkcjach agregujących. Do tego służą mechanizmy:

- **operatory logiczne**, to odpowiednio: suma i iloczyn logiczny, wymagające argumentów logicznych.:

&& - oznaczający koniunkcję „i”

|| - oznaczający alternatywę „lub”

na przykład :

waluty równe PLN i EUR wybiera warunek:

$\$waluta = PLN \ \&\& \ \$waluta = EUR$

waluty równe USD lub EUR wybiera warunek:

$\$waluta = USD \ || \ \$waluta = EUR$

dodanie jednoczesnego warunku na saldo wygląda następująco:

$(waluta = USD \ || \ \$waluta = EUR) \ \&\& \ \$saldo > 0$

i wybierze walutę USD lub EUR, których saldo jest większe od zera

- **operatory porównania liczb**, to:

= oznaczający: **równy**

!= oznaczający: **różny**

< oznaczający: **mniejszy**

<= oznaczający: **mniejszy lub równy**

> oznaczający: **większy**

>= oznaczający: **większy lub równy**

- **operatory porównania wartości tekstowych:**

= oznaczający: **równy**

!= oznaczający: **różny**

Wartości tekstowe można porównywać stosując wyrażenia regularne (opisane w dalszej części rozdziału).

Operator ‘równe’ i ‘różne’ wykonywany jest na argumentach dowolnego typu, natomiast pozostałe operatory porównania wspierają typ liczbowy i daty.

- **porównywanie korzystające z operatora zbioru:** [<element1>, <element2>, ...] (opisane w dalszej części rozdziału).

Dyskryminator może być dowolnie długi i korzystać z wielu kolumn – wymaganie prawidłowego zachowania warunku może wymagać wykorzystania nawiasów.

Operacje są grupowane według priorytetów (rosnąco):

- "||"
- "&&"
- "=", "!=" , "<" , "<=" , ">" , ">="
- "+", "-"
- "*", "/"

6.2.6.3 Operator zbioru

Operator zbioru może występować jedynie po jednej stronie porównania i wspiera jedynie operacje:

= oznacza **należenie** do zbioru,

!= oznacza **nienależenie** do zbioru,

którego elementy są wymienione pomiędzy nawiasami [] i oddzielone przecinkiem.

Operacja równości (mnogościowe: 'należy') jest semantycznie równoważna alternatywie porównań wartości z każdym elementem zbioru. Jednocześnie porównanie do zbioru pustego jest zawsze nieprawdziwe.

Operacja nierówności (mnogościowe: 'nie należy') jest semantycznie równoważna koniunkcji porównań (operatorem "!=") do każdego elementu zbioru. Jednocześnie porównanie do zbioru pustego jest zawsze prawdziwe.

Operator ten służy uproszczonej notacji długich warunków o regularnej postaci (na przykład wymienianie dozwolonych wartości dla danej kolumny).

Przykładowo:

```
sum( $saldo, $waluta = PLN || $waluta = EUR || $waluta = USD || $waluta = GBP )
```

jest równoważne:

```
sum( $saldo, $waluta = [PLN, EUR, USD, GBP] )
```

6.2.6.4 Funkcje manipulujące na wartościach

- **toNumber** – jawna konwersja na typ liczbowy zmiennoprzecinkowy. Argument konwersji jest typu tekstowego. Drugi opcjonalny argument tekstowy to format, który w przypadku typu liczbowego

jest jednoznakowym separatorem dziesiętnym. W przypadku jego braku wybierany jest domyślny ('.').

Funkcja konieczna, gdy:

- o w źródle danych wartości mają inny niż wymagany (kropka) separator dziesiętny. Dla przecinka można to zapisać:

```
sum( to_number( $saldo, ';' )
```

- o nie wiadomo czy porównywać wartości tekstowe czy liczbowe (nie można tego wywnioskować z samego porównywania – wtedy wartości są traktowane jako teksty)

```
avg( $saldo1, to_number( $saldo1 ) = to_number( $saldo2 ) )
```

- **toBool** – jawna konwersja na typ logiczny. Argument konwersji jest typu tekstowego. Drugi opcjonalny argument tekstowy to format, który w przypadku typu logicznego jest postaci: '<wartość-prawda>|<wartość-nieprawda>' W przypadku jego braku przyjmowany jest domyślny ('true|false'),

na przykład:

```
sum( $saldo, to_bool( $interesujace, 'TAK|NIE' ) )
```

- **toDate** – jawna konwersja na typ daty. Argument konwersji jest typu tekstowego. Drugi opcjonalny argument tekstowy to format daty. W przypadku jego braku przyjmowany jest domyślny ('yyyy-MM-dd'),

Daty wykorzystywane są jako wartość mapowana zwracana do komórki:

```
to_date( $data_transakcji )
```

lub w przypadku, gdy data ma inny niż domyślny format:

```
to_date( $data_transakcji, 'dd.MM.yyyy' ) dla dat typu '1.01.2009'
```

jak również dotyczy to porównań, na przykład, gdy interesująca jest suma wartości transakcji od początku roku:

```
sum( $wartosc, to_date( $data_transakcji ) >= to_date( '2009-1-1' ) )
```

- **toString** – funkcja odwrotna do jawnych konwersji - z danego typu tworzy tekst. Pierwszy argument to przetwarzana na napis wartość, drugi to opcjonalny format zgodny z tymi z konwersji. W przypadku braku drugiego argumentu przyjmowany jest domyślny format,
- **abs** – wartość bezwzględna. Funkcja jest jednoargumentowa, przyjmuje i zwraca wartość typu liczbowego. Jako argumentu można użyć dowolnego wyrażenia arytmetycznego lub pojedynczej kolumny,

Przykładowo:

ilość rekordów z saldami przekraczających wartość progową, można zapisać:

```
count( *, abs( $saldo_ma - $saldo_winien ) > 10000 )
```

Wartości bezwzględnej można też użyć w przypadku niepewności z jakim znakiem (dodatnim czy ujemnym) dane występują w źródle danych:

```
sum( $saldo_ma - abs( $saldo_winien ) )
```

- **negate** – wartość przeciwna. Funkcja jest jednoargumentowa, przyjmuje i zwraca wartość typu liczbowego;
- **roundInt** – zaokrągla wartość zmiennoprzecinkową na całkowitoliczbową zgodnie z ogólnymi zasadami zaokrąglania. Funkcja jest jednoargumentowa, przyjmuje i zwraca wartość typu liczbowego;
- **toInt** – rzutowanie na liczbę całkowitą. W przypadku gdy liczba posiada niezerowe rozwinięcie dziesiętne zostanie zgłoszony błąd. Funkcja jest jednoargumentowa, przyjmuje i zwraca wartość typu liczbowego;
- **isInt** – sprawdzenie czy string jest reprezentacją liczby całkowitej. Funkcja jest jednoargumentowa i przyjmuje argument typu tekstowego, a zwraca wartość logiczną;
- **toUpper** – konwersja na wielkie litery. Funkcja jest jednoargumentowa i przyjmuje argument typu tekstowego;
- **toLowerCase** – konwersja na małe litery. Funkcja jest jednoargumentowa i przyjmuje argument typu tekstowego.

Funkcje **toLowerCase** i **toUpperCase** – przydatne są, gdy dane w źródle danych są niejednolite.

Przykładowo:

gdy w kolumnie \$waluta występują wartości 'pln' i 'PLN' sumując takie wartości można zapisać:

```
sum( $saldo, to_upper( $waluta ) = PLN )
```

- **trim** – obcięcie znaków pustych z początku i końca napisu. Funkcja jest jednoargumentowa i przyjmuje argument typu tekstowego;
- **reverse** – odwrócenie napisu. Funkcja jest jednoargumentowa i przyjmuje argument typu tekstowego;
- **substring** – obcięcie napisu. Funkcja jest trzy lub dwu argumentowa. Pierwszym argumentem jest napis będący podstawą wycinania. Pozostałe dwa argumenty to liczby całkowite określające odpowiednio indeks początku i końca nowego napisu będącego wynikiem. Długość tak uzyskanego napisu to 'indeks końca' minus 'indeks początku'. W przypadku niepodanie trzeciego argumentu funkcja zachowuje się tak jakby podana została długość napisu,

- **indexOf** – poszukuje podnapisu w podanym napisie. Dwa pierwsze argumenty to odpowiednio napis podstawowy i szukany podnapis. Trzeci argument jest opcjonalny i oznacza miejsce rozpoczęcia wyszukiwania. W razie braku przyjmowane jest 0. Zwracana wartość jest typu całkowitoliczbowego i oznacza indeks znaku, od którego zaczyna się znaleziony podnapis lub -1 w przypadku nieznaalezienia;
- **toNumber** – zamienia wskazany napis na liczbę (wspierane separatory dziesiętne to ',' oraz '.'). Funkcja jednoargumentowa,
- **dayDiff** – 3 argumentowa funkcja, w której 3 argument jest opcjonalny. Funkcja zwraca różnicę pomiędzy datami stworzonymi na podstawie 2 pierwszych argumentów. Domyślny format daty to yyyy-MM-dd. Dodatkowo użytkownik może za pomocą 3 argumentu zdefiniować format dat,
- **lastIndexOf** – poszukuje podnapisu w podanym napisie. Dwa pierwsze argumenty to odpowiednio napis podstawowy i szukany podnapis. Wyszukiwanie odbywa się od końca. Trzeci argument jest opcjonalny i oznacza miejsce rozpoczęcia wyszukiwania. W razie braku przyjmowana jest długość napisu. Zwracana wartość jest typu całkowitoliczbowego i oznacza indeks znaku, od którego zaczyna się znaleziony podnapis lub -1 w przypadku nieznaalezienia;
- **length** – długość napisu. Funkcja jest jednoargumentowa i przyjmuje argument typu tekstowego, a zwraca wartość całkowitą;
- **equalsIgnoreCase** – równość napisów z pominięciem wielkości liter. Zwraca wartość logiczną zgodną z wynikiem porównania dwóch napisów,
- **startsWith** – sprawdzenie czy napis jest prefiksem innego. Funkcja jest dwu argumentowa. Zwraca jest wartość logiczną informującą czy drugi argument jest początkiem pierwszego,
- **getDay** – zwraca dzień z daty. Funkcja jest jednoargumentowa,
- **getMonth** – zwraca miesiąc z daty. Funkcja jest jednoargumentowa,
- **getYear** – zwraca rok z daty. Funkcja jest jednoargumentowa,
- **addDay** – dodaje dni do daty. Funkcja jest dwu argumentowa – pierwszy to data stanowiąca podstawę modyfikacji a drugi to liczba całkowita wyznaczająca ilość dni. Ujemna ilość dni oznacza zmniejszanie daty. Funkcja zwraca datę powiększoną lub pomniejszoną o określoną ilość dni,
- **addMonth** – dodaje miesiące do daty. Funkcja jest dwu argumentowa – pierwszy to data stanowiąca podstawę modyfikacji a drugi to liczba całkowita wyznaczająca ilość miesięcy. Ujemna ilość miesięcy oznacza zmniejszanie daty. Funkcja zwraca datę powiększoną lub pomniejszoną o określoną ilość miesięcy,

- **addYear** – dodaje lata do daty. Funkcja jest dwu argumentowa – pierwszy to data stanowiąca podstawę modyfikacji a drugi to liczba całkowita wyznaczająca ilość lat. Ujemna ilość lat oznacza zmniejszanie daty. Funkcja zwraca datę powiększoną lub pomniejszoną o określoną ilość lat,
- **date** – zwraca datę. Funkcja przyjmuje trzy argumenty całkowitoliczbowe oznaczające odpowiednio dzień, miesiąc i rok;
- **diffDay** – odległość między datami w dniach. Funkcja jest dwu argumentowa – przyjmuje dwie daty i zwraca odległość liczoną w dniach. Zwracana wartość jest liczbą całkowitą nieujemną, jeśli pierwsza data jest mniejsza lub równa od drugiej lub ujemną w przeciwnym wypadku;
- **isLeapYear** – rok przestępny. Sprawdza czy rok jest przestępny dla podanej daty lub wartości całkowitej reprezentującej rok. Zwraca wartość logiczną,
- **case** – funkcja wyboru. Funkcja jest 3 argumentowa, pierwszy argument jest typu logicznego, pozostałe dwa są nieokreślone, muszą jednak być tego samego typu i taki typ przybiera wartość zwracana. Jeżeli pierwszy argument jest prawdziwy to funkcja przyjmuje wartość drugiego argumentu, w przeciwnym przypadku trzeciego.

Funkcja jest pomocna w przypadku, gdy należy w zależności od pewnych wyliczeń zdecydować skąd pobrać dane.

Przykładowo:

gdy do sumy sald wchodzi inna kolumna w zależności od typu rekordu:

```
sum( choose( $typ = A, $saldo1, $saldo2 ), $typ = [A, B] )
```

wówczas, wyrażenie sumuje rekordy dodając wartość \$saldo1 dla typu A i \$saldo2 dla typu B

- **concat** – konkatencja napisów. Funkcja przyjmuje dwa lub więcej napisów, które zostaną połączone w jeden.

6.2.6.5 Funkcje kontekstowe

Funkcje kontekstowe zwracają wartości związane z kontekstem aktualnego przetwarzania (w czasie zasilania).

- **getReportDate** – zwraca datę sprawozdawczą dla zasilanego sprawozdania (ostatni dzień miesiąca sprawozdania). Funkcja jest bez argumentowa.
- **getSystemDate** – zwraca aktualną datę. Funkcja jest bez argumentowa.

- **getCVNumber** – dostęp do zmiennej katalogowej o podanej nazwie (dopuszczalny tylko literał) o typie liczbowym.
- **getCVString** – dostęp do zmiennej katalogowej o podanej nazwie (dopuszczalny tylko literał) o typie tekstowym.
- **getCVDate** – dostęp do zmiennej katalogowej o podanej nazwie (dopuszczalny tylko literał) o typie data.
- **getCVBool** – dostęp do zmiennej katalogowej o podanej nazwie (dopuszczalny tylko literał) o typie logicznym.

6.2.6.6 Funkcje agregujące

Funkcje agregujące obliczają wartość dla tabeli.

Pierwszy argument oznacza wyrażenie będące podstawą funkcji (obliczanym dla każdego wiersza), **drugi argument** jest warunkiem logicznym (dyskryminator), którego spełnienie wyznacza zbiór wierszy będących podstawą funkcji agregującej.

Funkcje agregujące (z wyjątkiem funkcji **count**) zwracają wartość typu pierwszego argumentu. Pierwszy argument jest obowiązkowy.

- **sum** – suma wyrażeń tabelarycznych dla wszystkich wierszy wyznaczonych przez dyskryminator. Jeśli żaden wiersz nie spełnia warunku 0 jest zwracane. Pierwszy argument jest typu liczbowego,
- **avg** – średnia z wyrażeń tabelarycznych dla wszystkich wierszy wyznaczonych przez dyskryminator. Jeśli żaden wiersz nie spełnia warunku 0 jest zwracane. Pierwszy argument jest typu liczbowego,
- **same** – warunek jednej wartości, zwraca wartość wyrażenia tabelarycznego, jeśli dyskryminator wyznaczył wiersze o tej samej wartości wyrażenia. W przeciwnym wypadku błąd jest zgłaszany. Jeśli ani jeden wiersz nie spełnia warunku, również błąd jest zgłaszany. Pierwszy argument, wyznaczający wartość zwracaną może być dowolnego typu;
- **one** – warunek jednego wiersza, zwraca wartość wyrażenia tabelarycznego, jeśli dyskryminator wyznaczył jeden wiersz. W przeciwnym wypadku błąd jest zwracany. Pierwszy argument, wyznaczający wartość zwracaną może być dowolnego typu;
- **any** – wartość dowolnego wiersza spełniająca dyskryminator. Jeśli ani jeden wiersz nie spełnia warunku, błąd jest zgłaszany. Pierwszy argument, wyznaczający wartość zwracaną może być dowolnego typu;

- **first** – wartość pierwszego wiersza spełniająca dyskryminator. Jeśli ani jeden wiersz nie spełnia warunku, błąd jest zgłaszany. Pierwszy argument, wyznaczający wartość zwracaną może być dowolnego typu;
- **last** – wartość ostatniego wiersza spełniająca dyskryminator. Jeśli ani jeden wiersz nie spełnia warunku, błąd jest zgłaszany. Pierwszy argument, wyznaczający wartość zwracaną może być dowolnego typu;
- **min** – oblicza minimum dla wierszy wyznaczonych przez dyskryminator. Akceptowane typy to liczbowy i data. Jeśli żaden wiersz nie spełnia warunku błąd jest zgłaszany,
- **max** – oblicza maksimum dla wierszy wyznaczonych przez dyskryminator. Akceptowane typy to liczbowy i data. Jeśli żaden wiersz nie spełnia warunku błąd jest zgłaszany,
- **count** – oblicza ilość wierszy spełniających warunek wyznaczony przez dyskryminator. Pierwszym argumentem może być znak '*' co oznacza, że policzone zostaną wszystkie wiersze spełniające warunek, lub wyrażeniem dowolnego typu i wtedy zostaną policzone wystąpienia różnych wartości takiego wyrażenia.

Przykładowo:

`count(*, $saldo_winien >0)` - obliczy liczbę rekordów z saldem większym od zera

`count($typ_konta)` - obliczy liczbę typów kont występujących w zbiorze

Funkcja **same** może zostać użyta niejawnie, jeśli w wyrażeniu znajdują się kolumny niezwiązane z żadną inną funkcją agregującą.

W związku z tym wyrażenia:

$\$a+\b i **`same($a+$b, true)`** gdzie *a* i *b* to nazwy kolumn, są równoważne.

Część funkcji agregujących (min, max, sum i avg) jest podstawą tworzenia funkcji podwójnie agregujących.

Funkcje te spełniają kryteria funkcji agregujących dla wartości liczbowych rozszerzając listę argumentów o dowolną liczbę wyrażen grupujących.

Każde wyrażenie grupujące może być dowolnego typu a wektor wyznaczony przez złożenie wartości tych wyrażen stanowi o podzbiorze wierszy będących podstawą wewnętrznej funkcji agregującej.

Wartość zewnętrznej funkcji agregującej (będąca jednocześnie wartością wyznaczoną przez funkcję) obliczana jest na podstawie wartości dla każdego wektora argumentów grupujących.

Nazwa funkcji podwójnie agregującej powstaje przez złożenie dwóch funkcji agregujących oddzielonych znakiem '_' (np. 'sum_min').

6.2.6.7 Rachunek typów

Język generowany przez powyższą gramatykę podlega sprawdzeniu zgodności typów (zbiór poprawnych napisów jest zawężony o konieczność zgodności z typami wykonywanych operacji). Rachunek typów oparty jest na następujących zasadach.

- wszystkie literały (liczbowe, logiczne i tekstowe) mają określony typ.
- operacje arytmetyczne wymagają typu liczbowego operatorów, w przypadku, gdy typ argumentu nie jest określony wykonywana jest niejawną konwersją;
- operacje logiczne wymagają typu logicznego operatorów, w przypadku, gdy typ argumentu nie jest określony wykonywana jest niejawną konwersją;
- wszystkie operacje porównania wymagają zgodnych typów argumentów. Jeśli jeden typ jest nieokreślony zostaje wykonana niejawną konwersją. Jeśli oba typy są nieokreślone wybierany jest typ domyślny.
- operacje porównania '=' i '!=' mogą być zastosowane do wszystkich typów argumentów. Domyślny jest typ testowy.
- operacje porównania '<', '<=', '>' i '>=' mogą mieć argumenty typu liczbowego lub daty. Domyślny jest typ liczbowy.
- typy argumentów funkcji określone są przez ich dopuszczalne sygnatury. Wybranie sygnatury określa typ wartości zwracanej.
- w przypadku gdy funkcja posiada wiele możliwości typów dla danej ilości argumentów, wybierana jest ta wersja, która jest zgodna z typami argumentów, jeśli funkcja posiada nieokreślone typy rozstrzygających argumentów pozostaje nieokreślona;
- typy nieokreślone to kolumny lub funkcje, które na skutek nieokreślonych typów parametrów zwracają wartości uzależnione od wymaganego typu funkcji;
- konwersja typów nieokreślonych polega na zastosowaniu funkcji (**toNumber**, **toBool**, **toDate** z domyślnymi formatami) dla kolumn i konwersji niejednoznacznych argumentów dla funkcji;
- w przypadku gdy propagowanie zasad rozstrzygania typów nie doprowadzi do określenia typu kolumny – przyjmowany jest typ tekstowy,
- elementy zbioru muszą posiadać ten sam typ (automatyczna konwersja kolumn zachodzi podobnie do zwykłej operacji porównania). Typ argumentu porównania ze zbiorem musi być typu zgodnego z elementami zbioru,
- funkcje dostępu do zmiennych katalogowych zwracają wartość typu zmiennej.

6.2.6.8 Funkcje grupujące

Funkcje grupujące nie obliczają wartości, ale wyznaczają grupy wierszy, które są podstawą wyznaczania wartości innych wyrażeń znajdujących się w grupie. Istnieją dwie funkcje grupujące:

- **group** – grupowanie, przyjmuje niepustą listę wyrażeń grupujących dowolnego typu. Każdy wektor wartości argumentów wyznacza grupę.
- **conditionGroup** – grupowanie warunkowe jako pierwszy argument przyjmuje warunek, który spełniają wszystkie grupy wyznaczone przez funkcję. Kolejne argumenty to lista wyrażeń grupujących dowolnego typu. Każdy wektor wartości argumentów wyznacza grupę.

6.2.6.9 Automatyczne formatowanie

Edycja i walidacja wyrażenia wiążą się z automatycznym formatowaniem, które oprócz wyrównania tekstu wyrażenia może zmienić następujące elementy:

- usunąć nadmiarowe nawiasy,
- usunąć cudzysłów z literałów tekstowych, jeśli rozpoczynają się od litery, nie posiadają znaków białych i nie należą do słów zarezerwowanych (**true**, **false**, **not**, **if**, **then**, **else**, **let**, **in**);
- usunąć wielokrotne zagnieżdżenia operatora ‘-’ przed literałami liczbowymi,
- usunąć drugi argument funkcji agregującej, jeżeli jest literałem i ma wartość **true**;
- usunąć formalny zapis funkcji **same**, jeśli nie posiada dodatkowych warunków i nie jest zagnieżdżona w żadnej innej funkcji agregującej;
- usunąć nadmiarowe wystąpienia konwersji kolumn (w przypadku, kiedy typ wyrażenia jest jednoznaczny i nie podano niedomyślnego formatu),
- zamienić argumenty porównania tak, aby po lewej stronie znajdowała się kolumna;
- zamienić argumenty porównania tak, aby po prawej stronie znajdował się operator zbioru.

6.2.6.10 Wyrażenia regularne

Wyrażenia regularne wykorzystywane są przy porównaniach kolumny z wartością tekstową.

Jeśli w napisie występują znaki ‘?’ , ‘*’ , ‘[’ lub ‘]’ , które nie są poprzedzone znakiem ‘\’ są traktowane jako **znaki specjalne**. Służą one do mniej precyzyjnego określenia wartości poszukiwanej kolumny.

? zastępuje dowolny (dokładnie jeden) znak,

* zastępuje dowolną ilość (również brak) znaków.

[] pomiędzy tymi nawiasami można wymienić znaki, z których dopasowywany będzie jeden znak napisu

Przykładowo:

$\$nazwisko = 'D*ski'$ oznacza nazwiska zaczynające się na literę 'D' i kończące na 'ski', natomiast

$\$konto = '11-[123]'$ będzie pasowało do numerów kont: '11-1', '11-2' i '11-3'.

6.2.6.11 Obliczanie wyrażeń tabelarycznych

Aby obliczyć wyrażenie tabelaryczne konieczne jest wczytanie zawartości źródła danych i sprawdzenie poprawności kolumn.

Jeśli wyrażenie wymaga innego niż tekstowy typu kolumny dokonywana jest odpowiednia konwersja. Możliwe jest zatem, że jedna kolumna zostanie poddana konwersji na różne typy w obrębie jednego wyrażenia w zależności od kontekstu występowania.

Kolumny (reprezentowane w wyrażeniu poprzez nazwy poprzedzone znakiem '\$') są zawsze łączone z najbardziej zagnieżdżoną funkcją agregującą której są argumentem:

$sum(one(\$a) + \$b)$ oznacza zatem, że kolumna *a* jest argumentem funkcji **one**, a *b* funkcji **sum**.

W przypadku zagnieżdżonych funkcji agregujących ich wartości obliczane są tylko raz.

Każdy poziom zagnieżdżenia wymaga niezależnego poziomu przetworzenia tabeli źródłowej.

Obliczanie wyrażeń objętych grupą odbywa się w dwóch etapach.

W pierwszym kroku wyznaczane są grupy na podstawie wyrażenia grupującego, a następnie w obrębie każdej grupy niezależnie obliczane są podległe jej wyrażenia.

Wartości warunków logicznych (głównie dyskryminatory) są obliczane tak szybko jak szybko znany jest wynik. W celu osiągnięcia większej wydajności zalecane jest zatem używanie warunków bardziej selektywnych na początku koniunkcji i mniej selektywnych na początku alternatywy.

6.2.6.12 Kolumny wirtualne

Tabelaryczne źródło danych może definiować kolumny wirtualne.

Wartość kolumny wirtualnej:

- wyrażana jest poprzez wyrażenia tabelarycznego źródła danych nieposiadających agregatów i dostępu do zmiennych katalogowych;
- obliczana jest dla każdego wiersza z wykorzystaniem innych (niewirtualnych) kolumn z wartościami dla tego wiersza,

- dopisywana jest do źródła danych i wykorzystywana w ten sam sposób co zwykłej kolumny w wyrażeniach mapujących. Symbol kolumny wirtualnej nie może pokrywać się z kolumnami źródła.

6.2.6.13 Śledzenie przetwarzania tabelarycznego źródła danych

Przetwarzanie tabelarycznego źródła danych polega na sekwencyjnym przejściu po wszystkich wierszach źródła danych.

Możliwe jest śledzenie wykorzystania poszczególnych wierszy na poziomie danego wyrażenia co oznacza, że można wybrać ze źródła wszystkie wiersze wykorzystane przy wartościowaniu danego wyrażenia.

Wiersz jest wykorzystany przez wyrażenie mapujące, jeśli spełnia warunek występujący w przynajmniej jednej funkcji grupującej pierwszego poziomu (niezagnieżdżonej).

Funkcjonalność ta jest w aplikacji wykorzystywana w dwóch przypadkach:

- weryfikacji mapowań – możliwość wyświetlenia wszystkich wierszy, które brały udział w obliczaniu wskazanej komórki – osobno dla każdego wykorzystywanego źródła danych;
- śledzenia nieużywanych rekordów – możliwość zapisu wszystkich wierszy, które nie zostały wykorzystane przez ani jedno wyrażenie mapujące z aktualnie obliczanego zbioru wyrażeń – osobno dla każdego wykorzystywanego źródła danych.

6.2.6.14 Źródła danych typu Excel komórkowy

Źródło danych umożliwia zasilanie z pliku(ów) Excel o dowolnej strukturze. Konstrukcja źródła danych polega na tworzeniu tabeli, gdzie każdy wiersz reprezentuje określoną komórkę z pliku. Źródło danych korzysta z tych samych mechanizmów co inne tabelaryczne źródła danych. Dodatkowy mechanizm umożliwia bezpośrednio adresowanie komórek pliku Excel za pomocą skrótu:

@nazwa_pliku!nazwa_arkusza!adres_komórki

gdzie nazwa pliku i nazwa arkusza są opcjonalne. Jednak wyrażenie zakłada, że w źródle jest co najwyżej jedna komórka o podanym adresie. Wyrażenie adresujące:

@file.xls!sheet1!A1

jest równoważne wyrażeniu:

```
oneOrNull( $sys.value, $sys.value != " && $sys.address = 'A1' && $sys.sheet_name = 'sheet1' && $sys.file_name = 'file.xls' )
```

6.2.7 Import i eksport wyrażeń mapujących

Dla każdego zbioru mapowań możliwe jest wykonanie eksportu wszystkich wyrażeń z wybranego podzbioru tabel. Wyrażenia takie są adresowane poszczególnymi faktami, więc można je zaimportować do innej taksonomii pod warunkiem, że określone fakty się w niej znajdują.

Źródła danych są współdzielone pomiędzy taksonomiami, dlatego nie są eksportowane. Wymagane jest jednak ich istnienie dla nowo importowanych wyrażeń.

Eksportowany format pliku musi być zgodny z formatem wybieranym podczas importu.

Format opisany jest poniższym XMLSchema:

```
<xs:schema id="fact-mappings" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="ds-context-type">
    <xs:sequence>
      <xs:element name="data-source" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:attribute name="ds-code" type="xs:string" use="required" />
          <xs:attribute name="ds-type" type="xs:string" use="required" />
          <xs:attribute name="ds-verion" type="xs:string" use="required" />
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="fact-mapping-type">
    <xs:sequence>
      <xs:element name="fact" type="xs:string" minOccurs="1" maxOccurs="1" />
      <xs:element name="expression" type="xs:string" minOccurs="1" maxOccurs="1" />
      <xs:element name="comment" type="xs:string" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="shared-fact-mapping-type">
    <xs:sequence>
      <xs:element name="fact" type="xs:string" minOccurs="1" maxOccurs="unbounded" />
      <xs:element name="expression" type="xs:string" minOccurs="1" maxOccurs="1" />
      <xs:element name="comment" type="xs:string" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="tuple-mapping-type">
    <xs:sequence>
      <xs:element name="tuple-item" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="group-expression" type="xs:string" minOccurs="0" maxOccurs="1" />
            <xs:element name="fact-mapping" type="fact-mapping-type" minOccurs="0" maxOccurs="unbounded" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required" />
  </xs:complexType>
  <xs:complexType name="mapping-context-type">
```

```

<xs:sequence>
  <xs:element name="typed-dimension-mapping" minOccurs="0" maxOccurs="unbounded">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="member" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="group-expression" type="xs:string" minOccurs="0" maxOccurs="1" />
              <xs:element name="member-expression" type="xs:string" minOccurs="1" maxOccurs="unbounded" />
            </xs:sequence>
            <xs:attribute name="id" type="xs:string" use="required" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="id" type="xs:string" />
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
<xs:element name="fact-mappings">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ds-context" type="ds-context-type" minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="fact-mapping" type="fact-mapping-type" minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="shared-fact-mapping" type="shared-fact-mapping-type" minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="tuple-mapping" type="tuple-mapping-type" minOccurs="0" maxOccurs="unbounded"></xs:element>
      <xs:element name="mapping-context" type="mapping-context-type" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
    <xs:attribute name="fact-mappings-version" type="xs:string" use="optional" />
  </xs:complexType>
</xs:element>

```

6.2.7.1 Opis elementów zawartych w XML Schema `</xs:schema>`

Element `<fact-mappings>`, to:

- element głównego poziomu zawierający mapowania, posiada opcjonalny atrybut opisujący wersję mapowań.

Element `<ds-context>`, to:

- element pozwalający na zadeklarowanie używanych źródeł w danym zestawie mapowań. Wszystkie źródła użyte w zestawie powinny zostać wymienione w kolejnych elementach tego typu. Rodzicem tego elementu jest `<facts-mapping>`.

Element `<fact-mapping>`, to:

- definicja elementu opisującego mapowanie dla pojedynczej komórki.

Rodzicem tego elementu jest `<facts-mapping>`.

W skład tego elementu wchodzi dwa podelementy:

- **Element** `<fact>`, pozwalający na wskazanie faktu z instancji XBRL.

Adres faktu w instancji XBRL budowany jest zgodnie z następującą notacją BNF:

```
<kod_taksonomii>;<kod miary>;<kod wymiarów>;;<znacznik okresu>;<wartość>
```

gdzie:

<kod_taksonomi> Unikalny identyfikator typu taksonomii. Identyfikator ten można zobaczyć po najechaniu na komórkę w sprawozdaniu. Przykładowe identyfikatory to: C – COREP CRD III, CITS – COREP NBP ITS, LQITS – Liquidy NBP ITS.

<kod miary> jest jednoznacznym określeniem miary wartości (ang. measure).

Rolę kodów miar oraz wymiarów pełnią polskie etykiety techniczne lub dla taksonomii niezawierających etykiet technicznych identyfikator XML z elementu definiującego dany koncept.

<kod wymiarów> jest jednoznacznym określeniem kombinacji wymiarów i ich składowych. W przypadku wartości raportowanych w kontekście wielu wymiarów, poszczególne kody oddzielane są przecinkiem:

```
<kod wymiaru>,<kod wymiaru>,<kod wymiaru> ...
```

<kod wymiaru> jest zbudowany z identyfikatora bądź polskiej etykiety technicznej wymiaru oraz znacznika składowej wymiaru, oddzielonych dwukropkiem:

```
<identyfikator/etykieta wymiaru>:<znacznik składowej wymiaru>
```

<znacznik składowej wymiaru> dla wymiarów predefiniowanych jest identyfikatorem XBRL lub polską etykietą techniczną składowej.

Dla wymiarów definiowanych przez użytkownika (typed) identyfikator budowany jest z aktualnej wartości składowej wymiaru wg następującego algorytmu:

- o dla składowych zbudowanych z wielu pól – ich zawartość jest łączona w jeden zapis, oddzielony znakiem podkreślenia;
- o w otrzymanym zapisie białe znaki oraz znaki średnika, przecinka, dwukropka, odwrotnego ukośnika, mniejszości, większości, cudzysłowu i apostrofa są zamieniane na znak podkreślenia;

<kod wymiarów> może być wartością pustą.

- o **Element <expression>** określający wartość mapowania dla wskazanego faktu.

Język mapowań został omówiony w poprzednich rozdziałach.

Element **<shared-fact-mapping>**, to:

- element pozwalający na zapisanie identycznego mapowania dla wielu komórek.

Działa podobnie jak `<fact-mapping>` jednak pozwala na wskazanie wielu faktów, dla których jest zdefiniowane dane wyrażenie.

Element `<tuple-mapping>`, to:

- element grupujący mapowania dla faktów z krotek (ang. Tuple), w skład, którego wchodzi dwa podelementy:
 - `<grouping-expression>` definiujący wartość funkcji grupującej dla krotek (ang. Tuple),
 - `<fact-mapping>` wyrażenie definiujące mapowanie dla konkretnego członka krotki (ang. Tuple).Sposób opisu tego elementu jest identyczny jak dla elementu `<fact-mapping>` będącego dzieckiem elementu `<fact-mappings>`.

Element `<mapping-context>`, to:

- element pozwalający na zdefiniowanie mapowań wymiarów użytkownika (ang. Typed Dimension). Podobnie jak dla `<tuple-mapping>` definiowana jest wartość funkcji grupującej. Dla wymiarów użytkownika, których członkowie zawierają więcej niż jeden element konieczne jest podanie mapowania dla każdego z nich.

6.3 Magazyn migawek

Z magazynem migawek połączone są następujące pojęcia:

Migawki, które budowane są w oparciu o wskazanie tabel oraz źródeł danych dla tych tabel.

Użytkownik ma możliwość wykorzystania utworzonych migawek w narzędziu ETL, które jest wbudowane w aplikację aSIST.

Definicja kolejnych przetwarzań ETL oraz tabel startowych, przechowywana jest w bazie danych aplikacji aSIST.

Instancja migawki - powstaje w momencie wykonania procesu zasilania danymi.

Dane instancji migawki przechowywane są w oddzielnej bazie danych DerbyDB.

Z uwagi na potencjalnie ogromną ilość takich danych, dla każdego okresu powstaje niezależna instancja bazy danych. Tabele powstające w bazie danych prefiksowane są wersją definicji oraz wersją danych.

Przechowywanie historycznych danych pozwala na uruchomienie procesu „weryfikacji mapowań” dla historycznych sprawozdań i definicji. Proces ten wykorzystywany jest do weryfikacji wartości znajdującej się na sprawozdaniu w oparciu o wyniki obliczeń na danych źródłowych.

6.4 ETL

ETL (ang. Extract, Transform, Load) to narzędzie, które służy do transformacji danych źródłowych przed użyciem ich w module mapowań.

Transformacja składa się z kroków definiowanych przez użytkownika w oknie edycji ETL.

Kroki pobierają dane z tabel zdefiniowanych w migawce. Każdy krok (oprócz wejściowych) używa danych z jednego lub więcej kroków wejściowych.

Większość kroków posiada kolumny oraz indeksy opisujące wynik przetwarzania dane.

6.4.1 Tabela wejściowa

Krok odpowiadający tabeli zdefiniowanej w migawce. Krok ten jest tylko do odczytu. Po zmianie definicji migawki kroki należy odświeżyć otwierając okno ETL i w miarę potrzeb zmienić definicję kroków zależnych od zmodyfikowanego kroku wejściowego.

6.4.2 Instrukcja SQL

Krok wykonujący zapytanie SQL podane przez użytkownika. Wynik zapytania jest zapisywany poprzez dodanie do zapytania prefiksu z poleceniem **insert**.

6.4.3 Skrypt wiersza

Skrypt w języku Groovy wykonywany dla każdego wiersza danych z kroku poprzedzającego. Na wejściu skrypt otrzymujemy mapę z zawartością przetwarzanego wiersza. W wyniku tworzymy mapę zgodną ze strukturą kolumn zdefiniowanego w aktualnym kroku.

6.4.4 Skrypt tabeli

Skrypt w języku Groovy wykonywany jest raz dla całej tabeli. Skrypt pozwala na wykonanie zapytań SQL i dowolną manipulację danych. Skrypt powinien sam uzupełnić dane aktualnego kroku.

6.4.5 Walidacja wiersza

Krok uruchamiający zestaw walidatorów sprawdzających poprawność wierszy za pomocą walidatorów:

- Niepusta kolumna,

- Skrypt Groovy.

6.4.6 Walidacja tabeli

Krok uruchamiający zestaw walidatorów sprawdzających poprawność tabel za pomocą walidatorów:

- Niepusta tabela,
- Skrypt Groovy,
- SQL.

6.5 Praca w środowisku wielostanowiskowym

W module mapowań możliwa jest jednoczesna praca wielu użytkowników na różnych stanowiskach aSIS, łączących się z jedną bazą danych.

Wymagana jest przy tym zawsze:

- baza Oracle.

W wersji wielostanowiskowej możliwa jest:

- praca wielu użytkowników **na różnych zbiorach oraz źródłach** i nie ma wówczas żadnych ograniczeń w ich edytowaniu.

Nie jest przy tym możliwe jednoczesne tworzenie zbiorów mapowań i źródeł oraz ich edytowanie przez wielu użytkowników.