

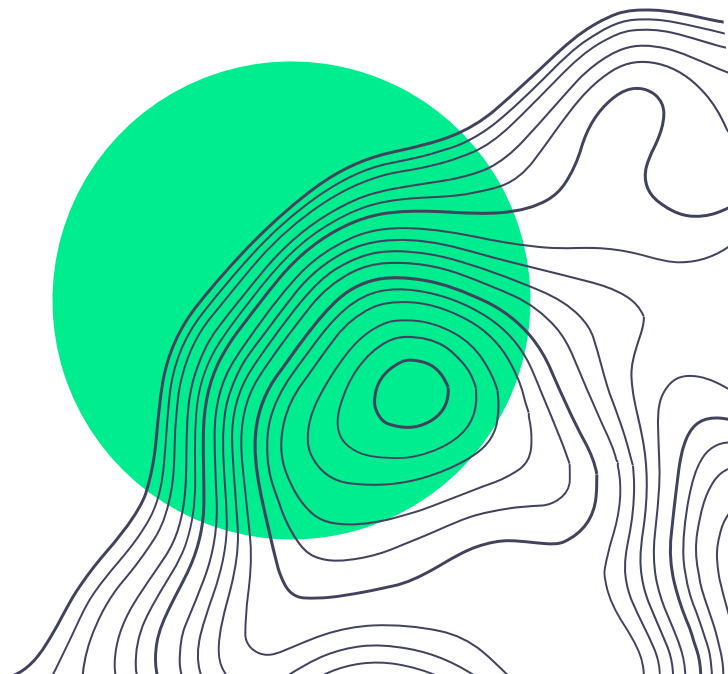
# Oprogramowanie integracyjne nowej generacji



Podręcznik

## SPIS TREŚCI

1. Kontekst
2. Cel oprogramowania GRAVITY
3. GRAVITY jako warstwa systemu informatycznego
4. Dla kogo oprogramowanie GRAVITY
5. Opis funkcjonalny środowiska GRAVITY
6. Wywołanie procesu GRAVITY
7. Rodzaje operatorów GRAVITY
8. Opis operatorów GRAVITY
9. Przykłady zastosowań



## 1. KONTEKST

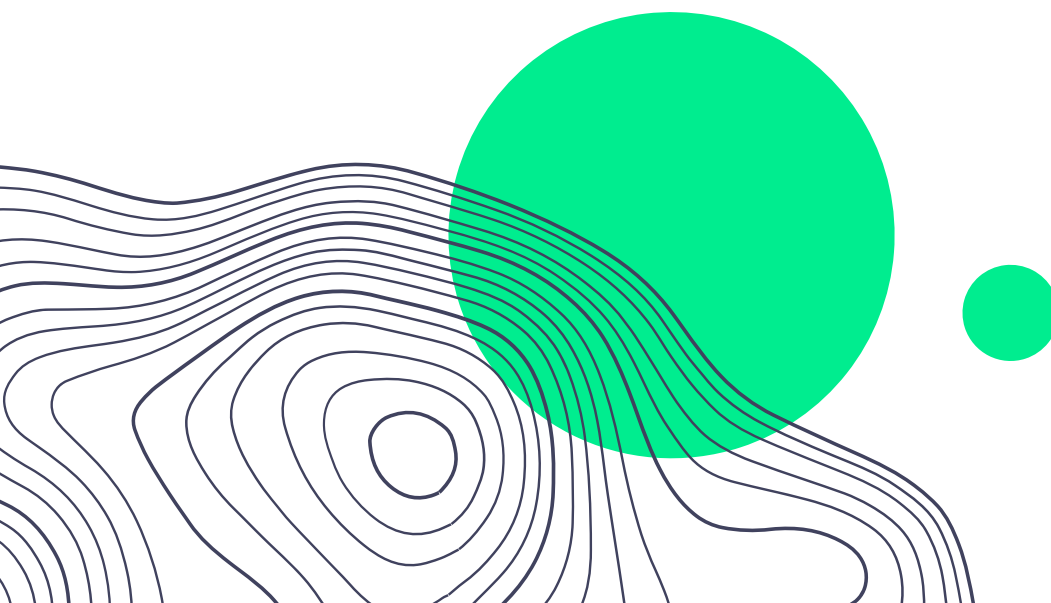
Wspomaganie informatyczne przedsiębiorstwa rozwija się w dwóch kierunkach: wzrasta zaawansowanie technologiczne oraz nieustająco zmienia się zakres i złożoność funkcjonalna oprogramowania.

Powyższa refleksja rodzi następujący problem: jak zwiększać zaangażowanie informatyki w prowadzenie przedsiębiorstwa w skalowalny sposób, nie zamrażając funkcjonujących rozwiązań, lecz w sposób elastyczny zareagować na nowe procesy oraz nowocześniejsze, bardziej wydajne technologie.

Nie mamy wątpliwości, że ciągła wymiana oprogramowania, tak jak to się dzieje w przypadku jednego oprogramowania obsługującego wszystkie obszary lub poleganie tylko na jednym dostawcy informatyki to droga niebezpieczna i mało efektywna: obok uzależnienia od jednego dostawcy, generuje ogromną pracę i środki finansowe.

Naszym zdaniem najefektywniejszą strategią informatyczną, a więc najbardziej skalowalną oraz najtańszą, jest stosowanie oprogramowania „otwartego” i wielowarstwowego, zdolnego do komunikacji z innymi rozwiązaniami dziedzinowymi dającymi się integrować w dowolny sposób. Dzięki temu efektywność stosowanego oprogramowania nie jest ograniczana do efektywności najsłabszego elementu, lecz jest sumą potencjału każdego elementu ustroju informatycznego.

Posłużmy się następującą metaforą: poszczególne rozwiązania informatyczne to obszary mózgu, które są odpowiedzialne za jakiś – czasami skromny – zakres „usług” lub funkcji, jednak połączone za pomocą neuronów (które tworzą układ nerwowy) mają potencjał wielokrotnie przekraczającą sumę potencjału każdego obszaru mózgu z osobna.




## 2. CEL OPROGRAMOWANIA GRAVITY

GRAVITY to oprogramowanie wspierające integrację różnych środowisk programistycznych. W środowisku GRAVITY możesz przygotować projekty ekstrakcji danych, transferu danych, integrować dane z różnych środowisk informatycznych, przetwarzać dane, standaryzować, walidować.

GRAVITY to środowisko informatyczne magistral danych, gdzie wykorzystując dostępne operatory można zamodelować proces przepływu danych i stworzyć mechanizmy przetwarzania, które są poza wyobraźnią także twórców oprogramowania GRAVITY.

Proponujemy, aby wyobrazić sobie GRAVITY jako wielką szynę danych, dowolnie zaprojektowaną, mogącą rozgałęziać się w każdym kierunku, po magistralach mkną terabajty danych we wszystkich kierunkach. Danych, które w trakcie transferu mogą być kontrolowane, walidowane, przetwarzane...

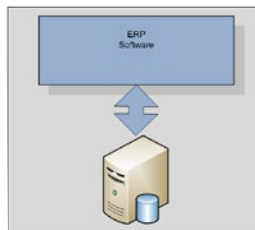


Odwołując się do wyobraźni czytelnika oraz metafory przytoczonej przez nas w rozdziale opisującym kontekst, jeżeli cały ustrój informatyczny przedsiębiorstwa jest „mózgiem”, to GRAVITY może stać się „neuronem” lub „siecią neuronów” transferującą informację pomiędzy obszarami „mózgu”....

## 3. GRAVITY JAKO WARSTWA SYSTEMU INFORMATYCZNEGO

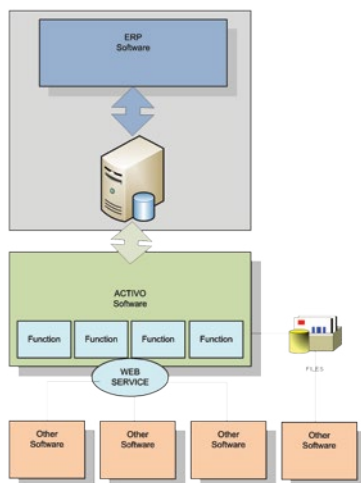
GRAVITY to środowisko ułatwiające rozwarstwienie złożonych systemów informatycznych. Dla zobrazowania wpływu rozwarstwienia systemu informatycznego na potencjał rozwojowy, prezentujemy poniższe diagramy.

### 3.1 DIAGRAM ROZWIĄZANIA Z MINIMALNYM POTENCJAŁEM ROZWOJU



- Oprogramowanie wraz z bazą danych stanowi całość (oprogramowanie zawiera całą logikę dostępu oraz obsługi danych)
- Wykorzystanie danych zawartych w bazie danych wiedzy tylko poprzez funkcje systemu ERP
- Najmniejsze rozszerzenie funkcji systemu ERP wymaga wymiany całego oprogramowania ERP
- Duże uzależnienie od jednego dostawcy informatyki (dostawcy systemu ERP)

### DIAGRAM ROZWIĄZANIA Z DUŻYM POTENCJAŁEM ROZWOJU



- Raz zbudowaną warstwę dostępu do danych można wykorzystać wielokrotnie do różnych zastosowań
- Rozwijając oprogramowanie specjalizowane będzie można skupić się na interfejsie; za ekstrakcję, walidację, transmisję danych będzie odpowiedzialna warstwa GRAVITY
- Następuje rozdział kompetencji: osoby o różnych kwalifikacjach będą zajmować się projektem w GRAVITY oraz projektem wytworzenia aplikacji dedykowanej

### KONKLUZJA



Jeżeli ustrój informatyczny przedsiębiorstwa jest wielowarstwowy, to wówczas jest mobilniejszy, sprawniejszy oraz bardziej rozwojowy od rozwiązań informatycznych z mniejszą ilością warstw.

## 4. DLA KOGO JEST OPROGRAMOWANIE GRAVITY

GRAVITY to środowisko informatyczne, dla:

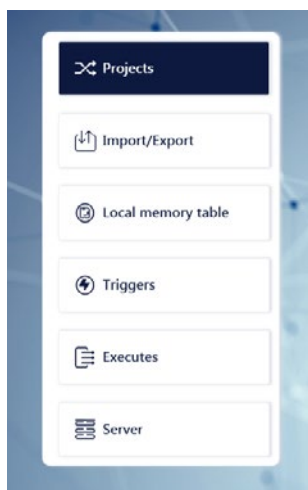
### **INTEGRATORÓW, FIRM I SŁUŻB INFORMATYCZNYCH**

- Twórcy oprogramowania nie muszą wytwarzać warstwy oprogramowania odpowiedzialnego za integrację (warstwa integracyjna może być zamodelowana w GRAVITY);
- Twórcy oprogramowania mogą raz wykonaną aplikację GRAVITY wykorzystać wielokrotnie w różnych przedsiębiorstwach;
- Zamodelowana w GRAVITY warstwa integracji umożliwi sprawne wymiany aplikacji front-end'u lub back-end'u;

### **PRZEDSIĘBIORSTW**

- Służby informatyczne mogą wykorzystać GRAVITY jako wewnętrzną szynę danych, łączącą ERP oraz rozwiązania dziedzinowe.

## 5. OPIS FUNKCJONALNY ŚRODOWISKA GRAVITY



Menu główne GRAVITY składa się z następujących opcji.

Poniżej przedstawiamy Tobie szczegółowy opis każdej opcji, w kontekście celu oraz zastosowania.



### 5.1 PROJECTS

W tej opcji utworzysz aplikację GRAVITY. Możesz zadeklarować dowolną liczbę projektów (aplikacji GRAVITY), dla różnych zastosowań. Poszczególne projekty mogą być niezależne, jednak istnieje możliwość powiązania ich ze sobą np. wynik jednego projektu może być na wejściu magistrali dla innego, zdefiniowanego przez Ciebie projektu GRAVITY.

Tworzenie projektu GRAVITY jest proste i intuicyjne. Wystarczy, że złapiesz operator i upuścisz go na pole robocze, a pomiędzy operatorami przeprowadzisz magistralę. Magistralę transferu danych możesz rozgałęziać, możesz mnożyć operatory i wyprowadzać przetworzone dane do wyznaczonych lokacji (baz, tabel, poczty etc.).

Zakres operatorów funkcjonalnych, którymi możesz posłużyć się podczas konfigurowania projektu GRAVITY jest przedstawiony w rozdziale OPIS OPERATORÓW GRAVITY.

#### 5.1.1 TWORZENIE NOWEGO PROJEKTU GRAVITY





Jeżeli chcesz stworzyć nową aplikację GRAVITY, uruchom opcję NEW – patrz ilustracja powyżej.



### 5.1.2 KASOWANIE PROJEKTU GRAVITY

New...

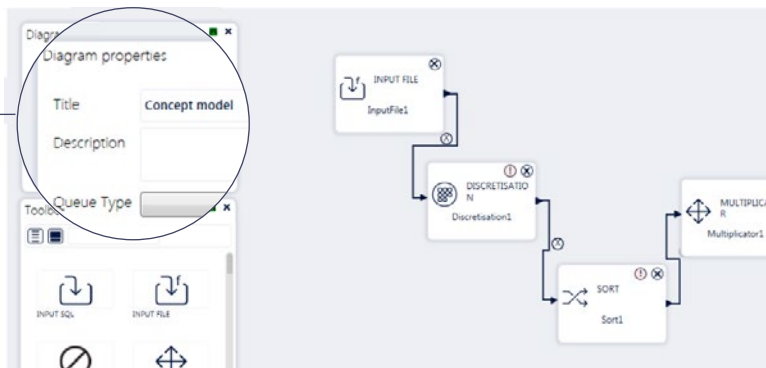
Jeżeli chcesz usunąć aplikację GRAVITY, uruchom opcję X – patrz ilustracja powyżej.

- Concept model 
- Concept Model 2 
- Concept Model 2\_1 
- Concept Model 30 

**i** Szczegółowy opis w zakresie kolejowania procesu znajdziesz w rozdziale **WYWOŁANIE PROCESU GRAVITY**.

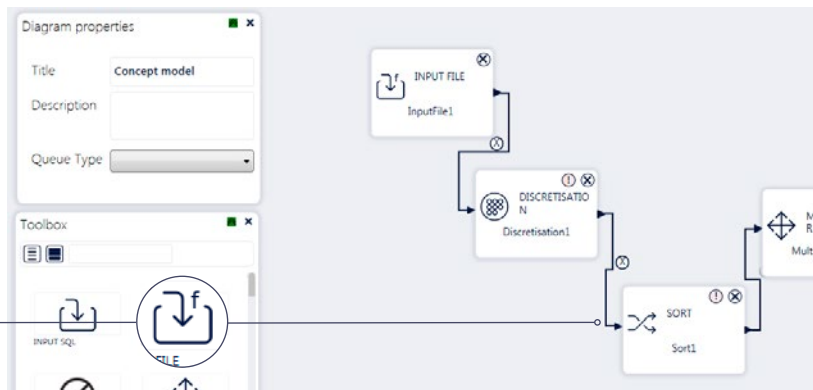
### 5.1.3 EDYCJA CECH APLIKACJI GRAVITY

W obszarze DIAGRAM PROPERTIES (patrz ilustracja powyżej) zadeklarujesz tytuł, opis projektu oraz sposób kolejowania uruchomionego procesu



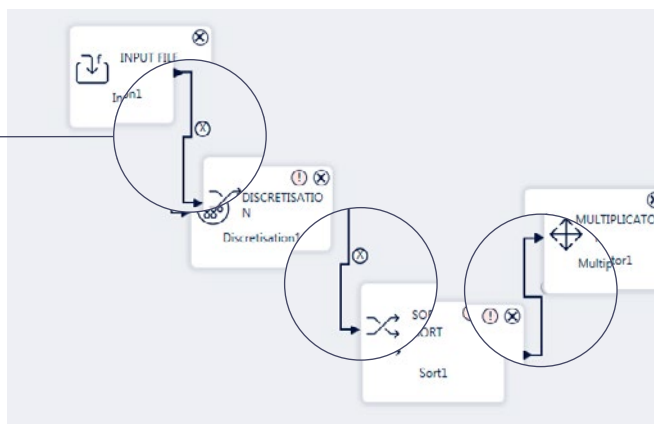
### 5.1.4 WYBÓR OPERATORA

Wybrany operator należy chwycić na obszarze TOOLBOX i upuścić na arkusz.



### 5.1.5 PRZEPROWADZENIE MAGISTRALI STRUMIENIA DANYCH

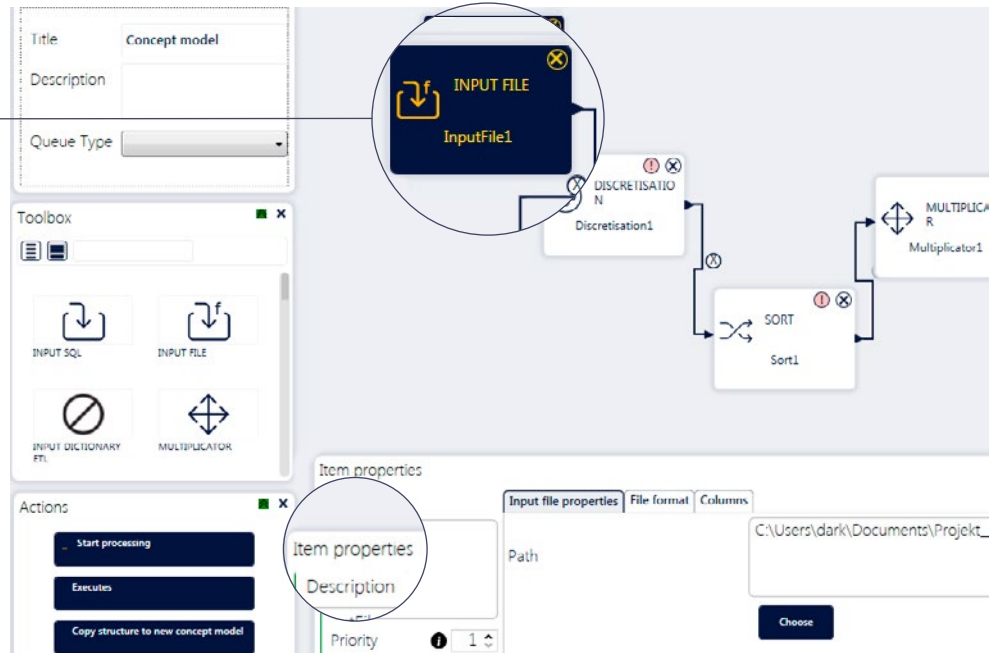
Magistralę rysujemy poprzez połączenie kursorem myszki wyjścia jednego operatora z wejściem operatora następnego.





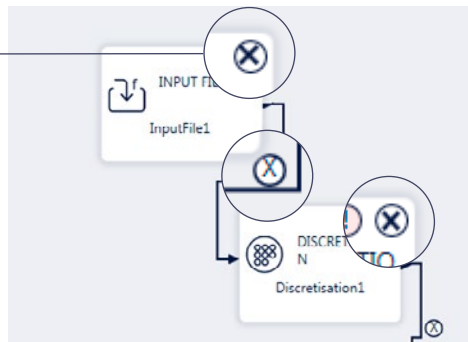
### 5.1.6 KONFIGURACJA OPERATORA

Każdy operator ma specyficzne tylko dla siebie parametry, które wypełniasz w obszarze ITEM PROPERTIES. Wybór operatora do konfiguracji następuje po kliknięciu kursorem myszy w obszar wybranego operatora.



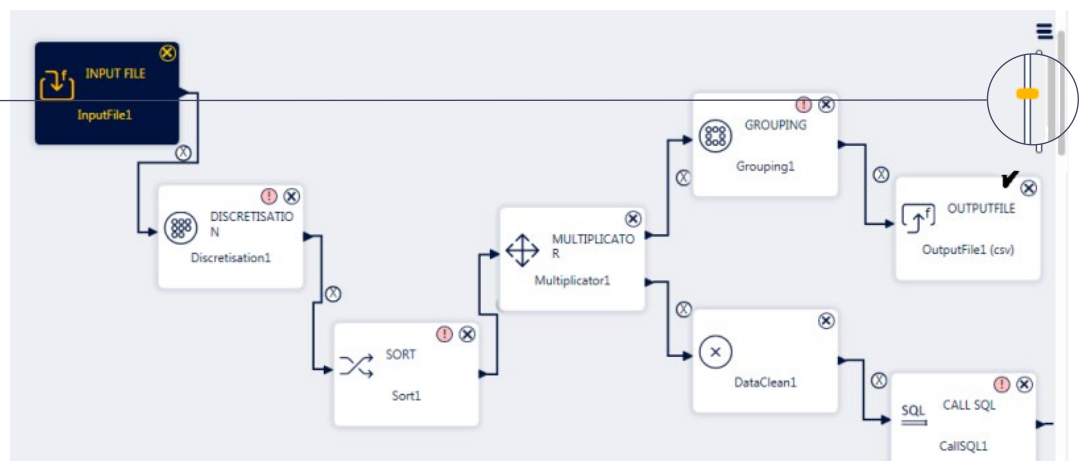
### 5.1.7 KASOWANIE OPERATORA I MAGISTRALI

Każdy operator oraz magistrala może zostać skasowana – wybór ikony kasowania (patrz ilustracja powyżej).



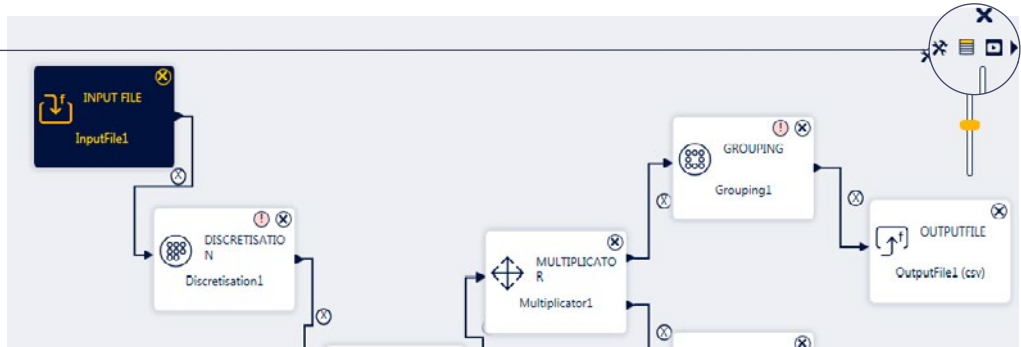
### 5.1.8 SKALOWANIE ARKUSZA

Dla komfortu przeglądania, arkusz możesz powiększać oraz pomniejszać (patrz ilustracja powyżej).



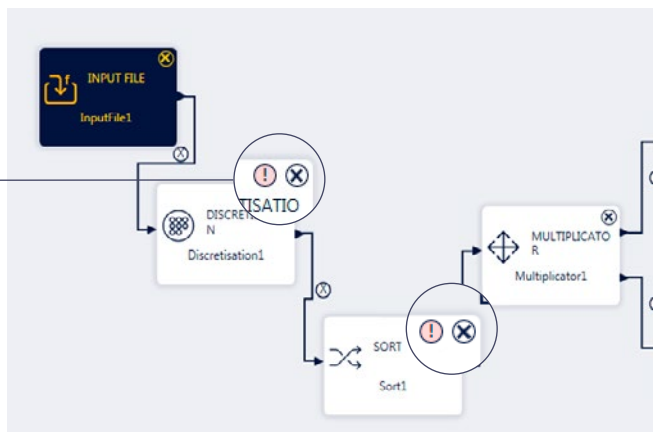
### 5.1.9 WYŁĄCZANIE OBSZARÓW ROBOCZYCH

Dla komfortu przeglądania, możesz wyłączyć poszczególne obszary robocze (DIAGRAM PROPERTIES, TOOLBOX, ACTIONS, ITEM PROPERTIES – patrz ilustracja powyżej).



### 5.1.10 SYGNALIZACJA NIEKOMPLETNOŚCI PROJEKTU

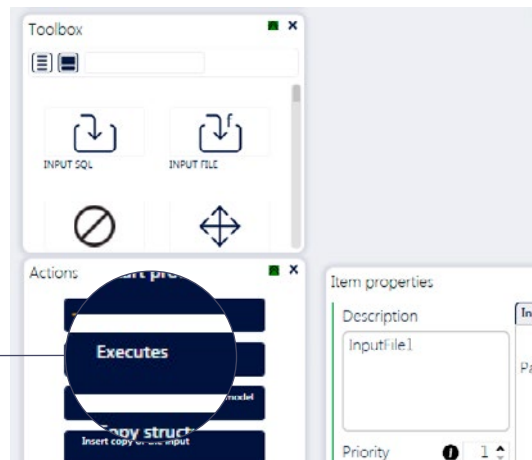
Jeżeli operator nie jest kompletnie skonfigurowany, na arkuszu pojawia się ikona sygnalizująca niekompletność (patrz ilustracja powyżej). Takiego projektu nie można uruchomić.



### 5.1.11 MANUALNE URUCHOMIENIE PROJEKTU

W obszarze ACTIONS (patrz ilustracja powyżej) masz możliwość manualnego uruchomienia aplikacji GRAVITY (szczegółowy opis w zakresie sposobów uruchamiania aplikacji GRAVITY znajdziesz w rozdziale **WYWOŁANIE PROCESU GRAVITY**).

### 5.1.12 PODGLĄD WSZYSTKICH URUCHOMIEŃ PROJEKTU GRAVITY

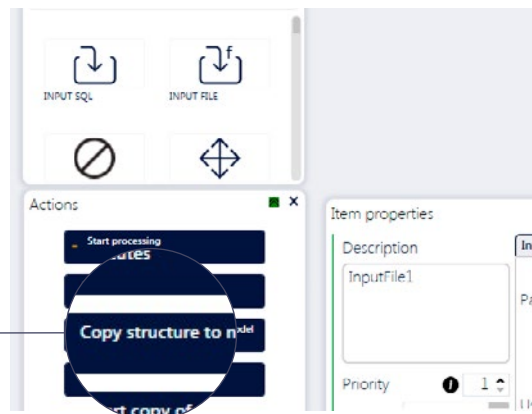


W obszarze **ACTIONS** masz możliwość dotarcia do kartoteki wszystkich uruchomień aplikacji



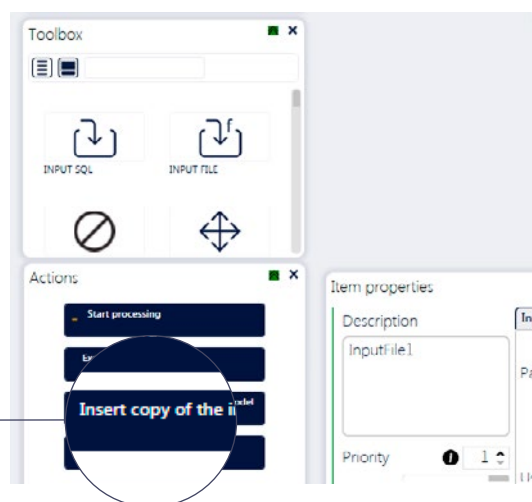
wraz z informacjami o czasie wywołania, rezultatach, ostrzeżeniach i błędach

### 5.1.13 KOPIOWANIE ARKUSZA PROJEKTU



W obszarze **ACTIONS** masz możliwość utworzenia nowej aplikacji GRAVITY poprzez skopiowanie aplikacji bieżącej.

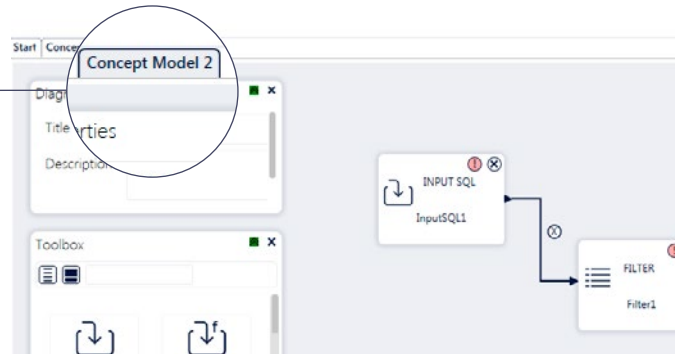
### 5.1.14 KOPIOWANIE SKONFIGUROWANEGO OPERATORA INPUT Z INNEGO ARKUSZA



W obszarze **ACTIONS** masz możliwość skopiowania operatora typu INPUT z wybranej innej aplikacji..


### 5.1.15 JEDNOCZESNY PODGLĄD KILKU PROJEKTÓW GRAVITY

Każdy projekt jest otwierany w nowej zakładce, a ty możesz szybko poruszać się pomiędzy projektami wybierając zakładki

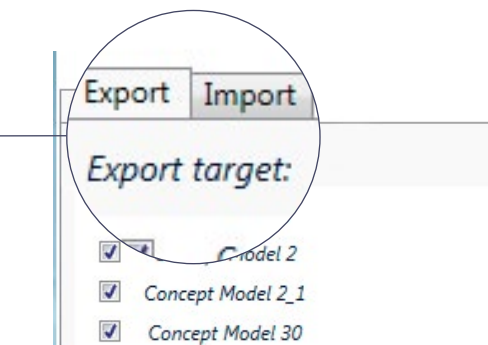


## 5.2 IMPORT/EXPORT

### 5.2.1 EXPORT

 W tej funkcji oprogramowania wyeksportujesz wszystko to co stworzyłeś oraz zaimportujesz to co stworzyli inni. Opcja umożliwia współpracę pomiędzy wieloma twórcami.

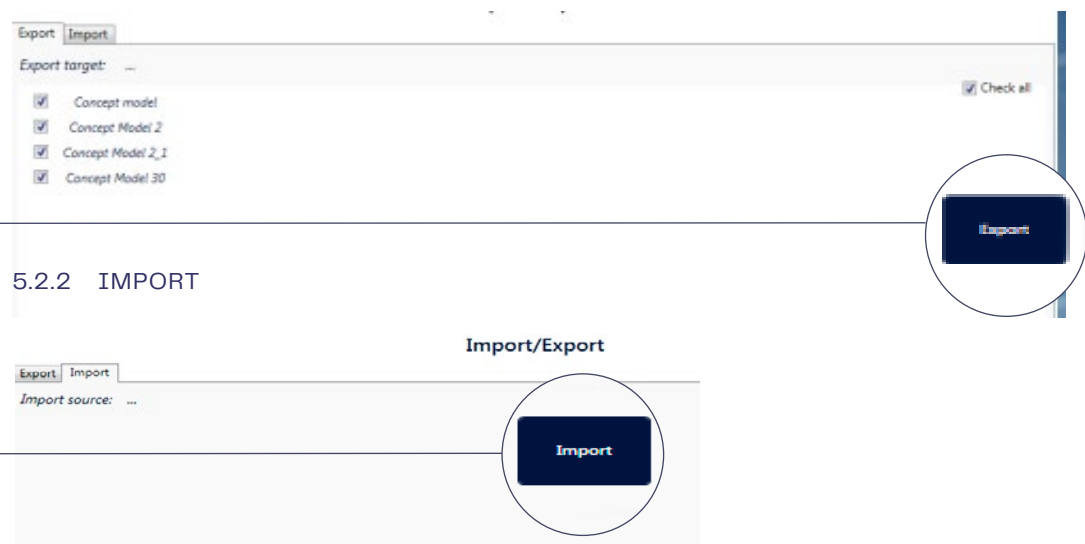
Pliki z zaznaczonymi do eksportu aplikacjami zostaną złożone w wybranej przez ciebie lokalizacji. Każdy plik ma format XML.



Akcję eksportu uruchamiasz opcją **EXPORT**

### 5.2.2 IMPORT

Jeżeli chcesz zaimportować aplikację GRAVITY wskaż plik XML zawierający definicję aplikacji, która np. wcześniej została wyeksportowana z GRAVITY) oraz uruchom opcję **IMPORT**.





## 5.3 LOCAL MEMORY TABLE

- i** Opcja, dzięki której zobaczysz dane, które zostały umieszczone w pamięci przez uruchomione aplikacje GRAVITY w trakcie bieżącej sesji (czyli w czasie od ostatniego uruchomienia środowiska GRAVITY). Dane te mogą być wykorzystywane przez systemy zewnętrzne, zaś dostęp do nich jest realizowany przez funkcję WEB SERVICE.



## 5.4 CALLS

W rozdziale [WYWOŁANIE PROCESU GRAVITY](#) opisaliśmy szczegółowo w jaki sposób możemy uruchomić proces (aplikację) GRAVITY.

Przypomnijmy wobec tego, że procesy GRAVITY możesz wywołać w następujący sposób:

- manualnie
- automatycznie przez inną aplikację GRAVITY
- automatycznie przez mechanizm wywołań czasowych
- automatycznie przez notyfikację bazy danych
- jako funkcja WEB SERVICE przez oprogramowanie zewnętrzne

W opcji CALLS możesz skonfigurować następujące mechanizmy wywoływania GRAVITY:

- automatycznie przez mechanizm wywołań czasowych
- automatycznie przez notyfikację bazy danych
- jako funkcja WEB SERVICE przez oprogramowanie zewnętrzne

Jeżeli chcesz więcej informacji o sposobach użycia powyższych opcji – przeczytaj rozdział [WYWOŁANIE PROCESU GRAVITY](#).



## 5.5 EXECUTES

Podgląd kartoteki wszystkich uruchomień aplikacji GRAVITY ze szczegółowym wskazaniem nazwy aplikacji, momentu uruchomienia, rezultatu oraz błędów i komunikatów.

## 5.6 SERVER

Serwer WEB SERVICE. W opcji CALLS opisaliśmy różne sposoby uruchamiania procesów (aplikacji) GRAVITY. Jednym ze sposobów jest wywołanie procesu GRAVITY jako funkcji WEB SERVICE poprzez akcję typu CALLS.

Zauważ, że możesz zdefiniować dowolną liczbę akcji CALLS, a więc możesz zdefiniować dowolną liczbę aplikacji GRAVITY i wywoływać przez oprogramowanie zewnętrzne jako funkcje WEB SERVICE.

De facto wszystkie te akcje przechodzą przez jeden serwer funkcji WEB SERVICE, który jest konfigurowany w omawianej opcji: musisz skonfigurować port, użytkownika oraz hasło.

Serwer jest uruchamiany i zatrzymywany poprzez opcje START/STOP \



## Server Logs

Date	Message	Client IP	Type
10/2/2020 10:17:15 AM	Protokół HTTP nie może zarejestrować adresu URL https://+9010/ActivoApi/. Używany		Error
10/2/2020 10:17:22 AM	Server is starting.		Message
10/2/2020 10:17:23 AM	Protokół HTTP nie może zarejestrować adresu URL https://+9010/ActivoApi/. Używany		Error

Wszystkie zdarzenia związane z działaniem serwera są rejestrowane i przedstawiane na liście



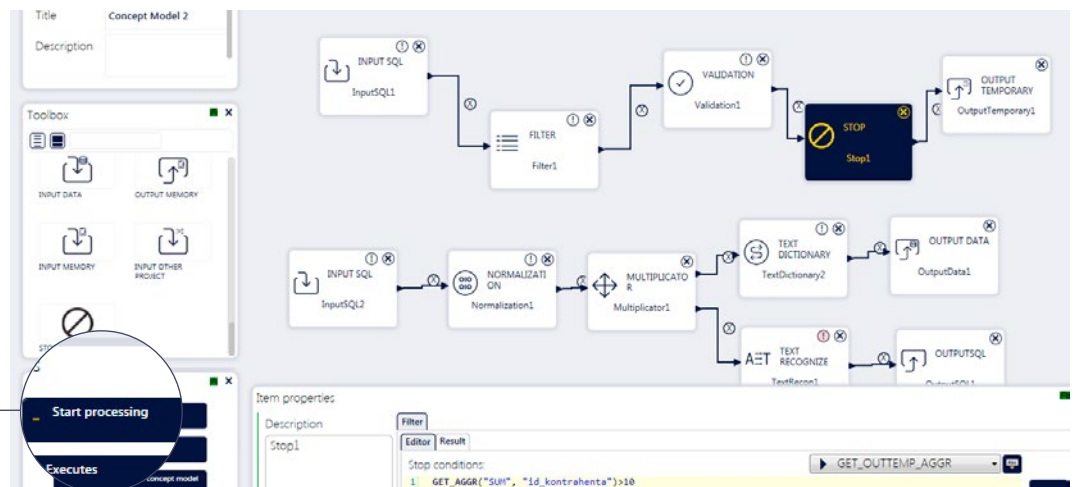
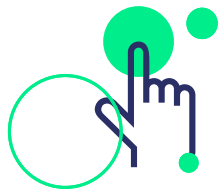
W kontekście wywołania funkcji WEB SERVICE prawdopodobnie pojawiło się u Ciebie pytanie: w jaki sposób oprogramowanie zewnętrzne może komunikować się z procesem GRAVITY?. Koniecznie przeczytaj szczegółowy opis dla operatorów INPUT DATA oraz OUTPUT DATA (rozdział [OPIS OPERATORÓW GRAVITY](#)). Te operatory są odpowiedzialne za transfer danych wejściowych oraz danych rezultatu.

## 6. WYWOŁANIE PROCESU GRAVITY

### SPOSOBY WYWOŁYWANIA PROCESU GRAVITY

- manualnie
- automatycznie przez inną aplikację GRAVITY
- automatycznie przez mechanizm wywołań czasowych
- automatycznie przez notyfikację bazy danych
- jako funkcja WEB SERVICE przez oprogramowanie zewnętrzne

### 6.1 WYWOŁANIE MANUALNE

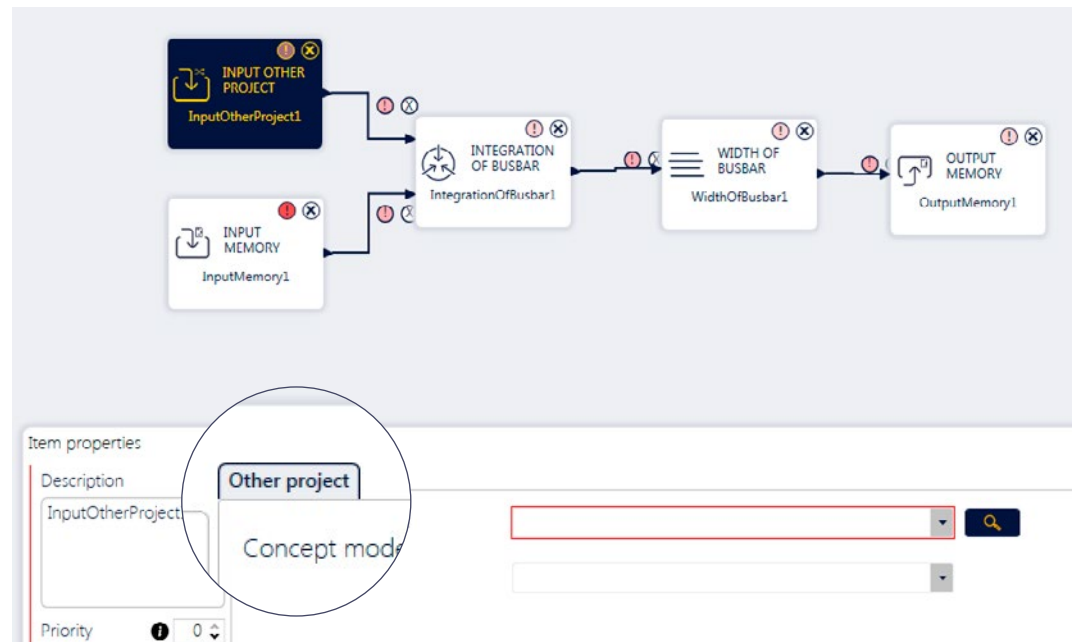


Na ilustracji wskazaliśmy opcję wywołującą proces, dostępną w oknie każdej aplikacji. Każda akcja przetwarzania jest zapamiętywana i dostępna do wglądu w opcji **EXECUTES**.



## 6.2 WYWOŁANIE AUTOMATYCZNE PRZEZ INNĄ APLIKACJĘ GRAVITY

Aplikacja GRAVITY może być wywołana przez inną aplikację GRAVITY. Poniżej na ilustracji masz przykład diagramu procesu, w którym zastosowano operator INPUT OTHER PROJECT (więcej informacji o parametrach tego operatora dostarczy Tobie opis operatora w rozdziale OPIS OPERATORÓW GRAVITY). Zastosowanie takiego operatora powoduje, że w pierwszej kolejności wywoływana jest inna aplikacja GRAVITY skojarzona (patrz parametry operatora) z tym operatorem. Wskutek zastosowania tej kontrolki i akcji wykonania skojarzonej aplikacji GRAVITY na wyjściu operatora INPUT OTHER PROJECT pojawi się strumień danych skojarzony z wybranym operatorem wyjścia (typu OUTPUT), w skojarzonej aplikacji wywołanej.



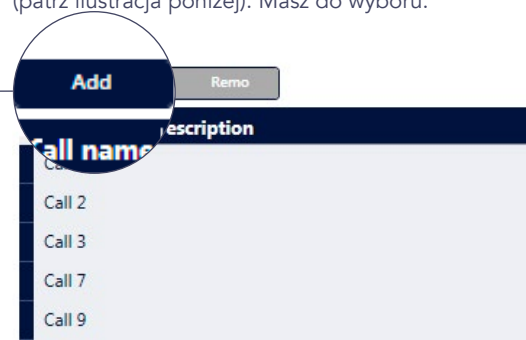
## 6.3 WYWOŁANIE AUTOMATYCZNE PRZEZ MECHANIZM WYWOŁAŃ CZASOWYCH

Jeżeli chcesz skonfigurować wywołanie aplikacji przez mechanizm wywołań czasowych, musisz wybrać w menu głównym opcję CALLS.

Decydując się na mechanizm wywołań czasowych, wybierasz ze słownika opcję SCHEDULER.

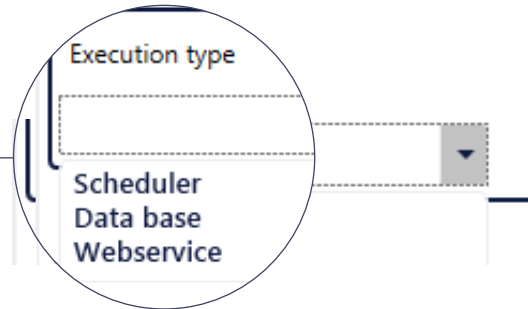
Aby dokończyć konfigurację, będziesz musiał jeszcze wskazać sposób identyfikowania czasu wywołania (patrz ilustracja poniżej). Masz do wyboru:

Wybierając opcję ADD możesz dodać nową konfigurację automatycznego uruchamiania aplikacji GRAVITY

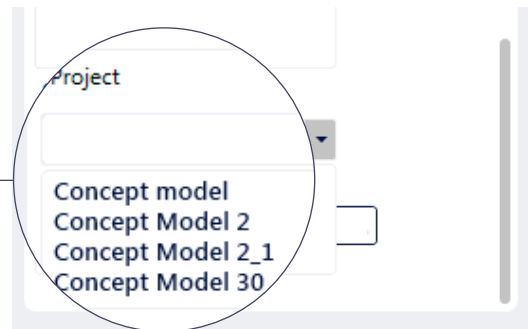




Po wyborze opcji ADD, musisz wybrać ze słownika rodzaj wywołania automatycznego



W obszarze PROPERTIES musisz wskazać również aplikację (projekt) GRAVITY, który odąd będzie skojarzony z konfigurowanym wywołaniem (patrz ilustracja poniżej).



- **Specific date**

Wywołanie odbędzie tylko raz we wskazanej dacie i wskazanym czasie

- **Every few minutes**

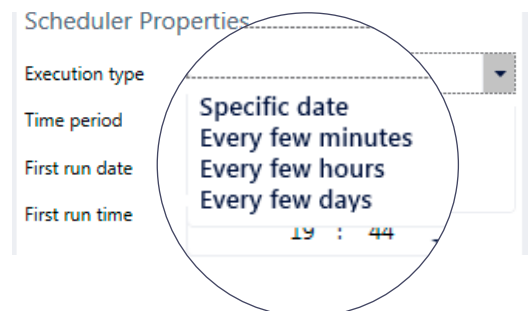
Wywołanie odbędzie się po raz pierwszy we wskazanej dacie i godzinie, a następnie będzie się powtarzać co zadeklarowaną liczbę minut.

- **Every few hours**

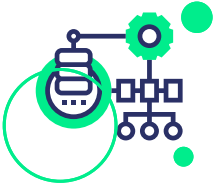
Wywołanie odbędzie się po raz pierwszy we wskazanej dacie i godzinie, a następnie będzie się powtarzać co zadeklarowaną liczbę godzin.

- **Every few days**

Wywołanie odbędzie się po raz pierwszy we wskazanej dacie i godzinie, a następnie będzie się powtarzać co zadeklarowaną liczbę dni.



Jeżeli masz skonfigurowaną akcję CALL typu SCHEDULER, oprogramowanie GRAVITY musisz pozostawić uruchomione, dzięki temu system wystartuje skojarzoną aplikację (proces) GRAVITY we wskazanym czasie lub interwałach czasowych.



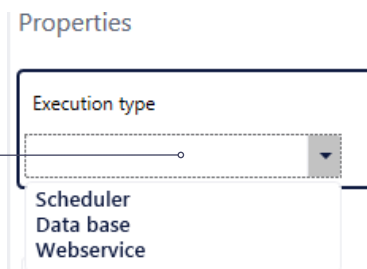
## 6.4 WYWOŁANIE AUTOMATYCZNE PRZEZ NOTYFIKACJĘ BAZY DANYCH

Jeżeli chcesz skonfigurować wywołanie aplikacji przez mechanizm notyfikacji bazy danych (inaczej: na podstawie sygnału pochodzącego z bazy danych), musisz wybrać w menu głównym opcję CALLS.

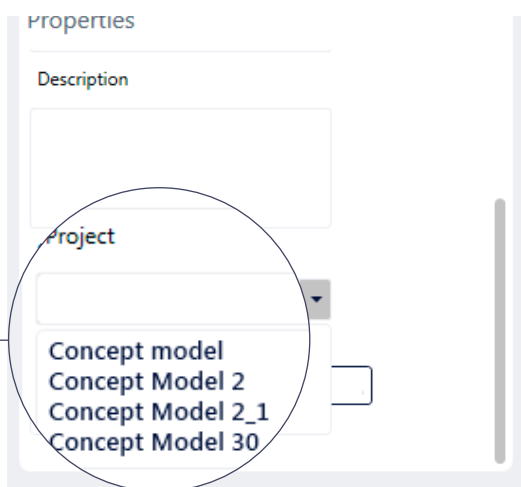
Wybierając opcję ADD możesz dodać nową konfigurację automatycznego uruchamiania aplikacji GRAVITY



Po wyborze opcji ADD, musisz wybrać ze słownika rodzaj wywołania automatycznego

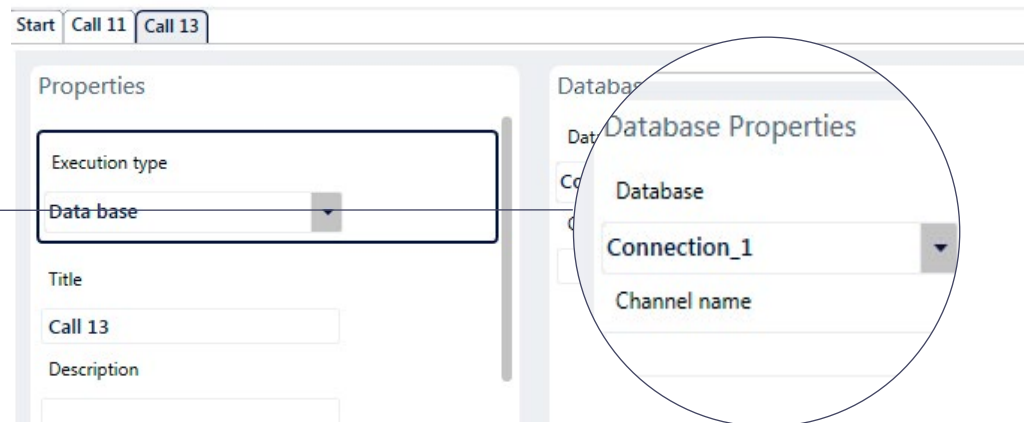


W obszarze PROPERTIES musisz wskazać również aplikację (projekt) GRAVITY, który odtąd będzie skojarzony z konfigurowanym wywołaniem



Decydując się na mechanizm notyfikacji bazy danych wybierasz ze słownika opcję DATEBASE.

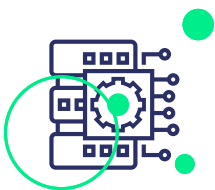
Aby dokończyć konfigurację będziesz musiał jeszcze wskazać połączenie z bazą danych (wybór ze słownika skonfigurowanych połączeń).



Decydując się na mechanizm wywołania poprzez notyfikację bazy danych wybierasz ze słownika opcję DATEBASE.

Aby dokończyć konfigurację, będziesz musiał jeszcze wskazać połączenie z bazą danych (wybór ze słownika skonfigurowanych połączeń – patrz ilustracja powyżej) oraz unikalną nazwę kanału nasłuchu bazy danych (kanał musi być wcześniej zadeklarowany w bazie danych i przypisany do jakiegoś zdarzenia, które wywołuje akcję notyfikacji bazy danych).

**i** Jeżeli masz skonfigurowaną akcję CALL typu DATEBASE, oprogramowanie GRAVITY musisz pozostawić uruchomioną. GRAVITY jest podłączone do bazy danych i nasłuchuje. Jeżeli we wskazanym kanale pojawi się sygnał (co oznacza zajście zdarzenia przypisanego w bazie danych do kanału), skojarzona aplikacja GRAVITY zostaje uruchomiona.

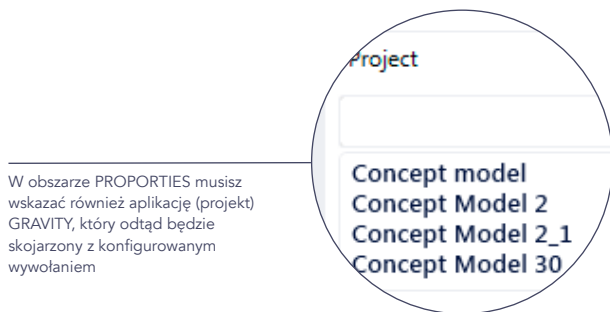


## 6.5 WYWOŁANIE JAKO FUNKCJA WEB SERVICE PRZEZ OPROGRAMOWANIE ZEWNĘTRZNE

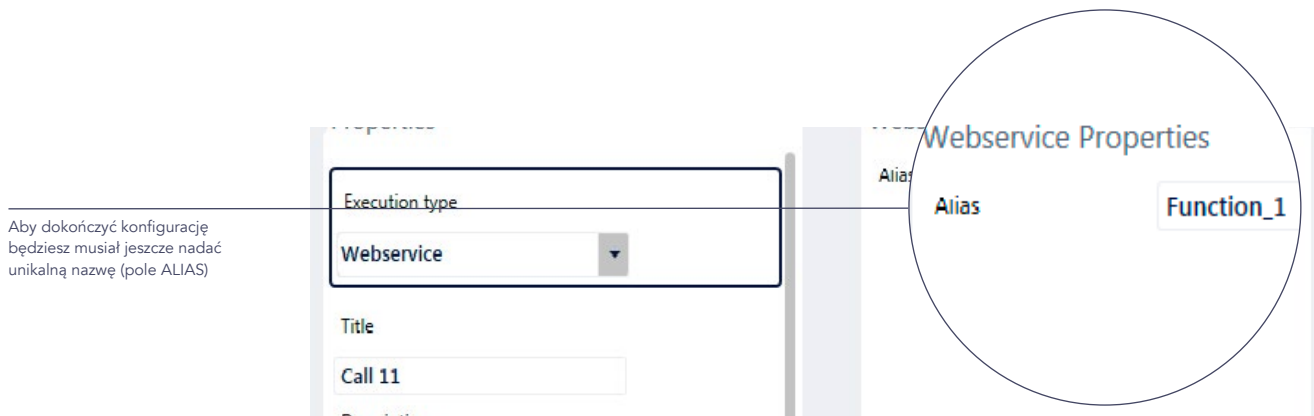
Jeżeli chcesz skonfigurować wywołanie aplikacji GRAVITY przez oprogramowanie zewnętrzne jako funkcji WEB SERVICE, musisz wybrać w menu głównym opcję CALLS.


Wybierając opcję ADD możesz dodać nową konfigurację automatycznego uruchamiania aplikacji GRAVITY





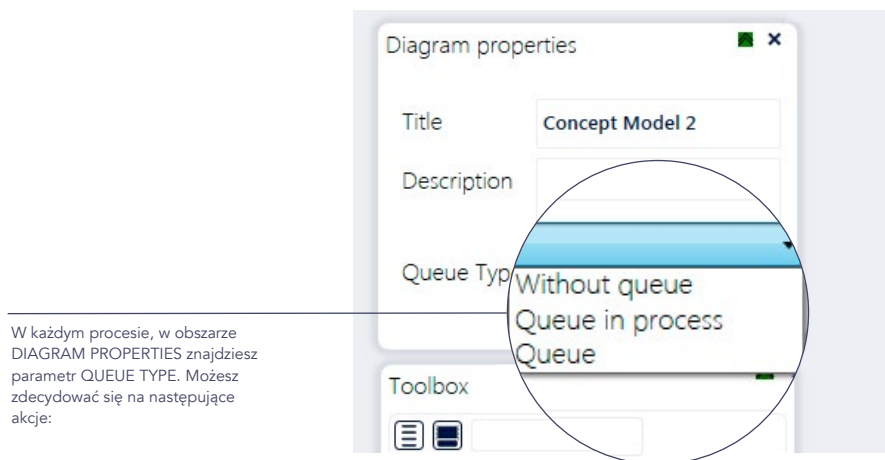
-  Decydując się na mechanizm wywołania aplikacji GRAVITY jako funkcji WEB SERVICE wybierasz ze słownika opcję WEBSERVICE.



-  Jeżeli masz skonfigurowaną akcję CALL typu WEBSERVICE, musisz jeszcze skonfigurować serwer WEB SERVICE (opcja SERVER). Poprzez ten serwer będą dostępne dla oprogramowania zewnętrznego wszystkie akcje CALL typu WEBSERVICE. Identyfikacja akcji następuje na podstawie unikalnej nazwy ALIAS. Wywołanie każdej akcji skutkuje uruchomieniem aplikacji (procesu) GRAVITY skojarzonej z akcją.

### 6.5.1 ZARZĄDZANIE KOLEJKAMI PROCESÓW GRAVITY

**i** Te same oraz różne aplikacje GRAVITY mogą być wielokrotnie wywoływane w krótkim odcinku czasu. Aby zarządzać tymi procesami, masz możliwość zadeklarowania sposobu kolejkwania zadań do przetworzenia.



W każdym procesie, w obszarze DIAGRAM PROPERTIES znajdziesz parametr QUEUE TYPE. Możesz zdecydować się na następujące akcje:

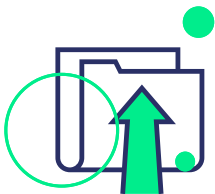
- Without queue.....proces wykonywany bezzwłocznie
- Queue in proces.....proces kolejkwany w ramach tych samych wywołań tej samej aplikacji
- Queue.....proces kolejkwany

## 7.

## RODZAJE OPERATORÓW GRAVITY

Postrzegaj aplikację GRAVITY jako mapę magistral łączących różne przetworniki, zwanych przez nas operatorami. GRAVITY posiada grupę przetworników (operatorów), które pozwalają na pozyskanie strumienia danych z zewnątrz. Strumień danych może potem krążyć po magistralach, być przetwarzany, łączony, rozpoznawany, czyszczony, rozgałęziany, utrzymywany w pamięci dla szybkiego dostępu, wysyłany na zewnątrz za pomocą specjalnych operatorów wyjściowych. Szerokość magistral oraz liczba operatorów użytych w aplikacji może być dowolna. Masz do dyspozycji również mechanizmy wyzwalania przepływów strumienia danych: od mechanizmów automatycznych do mechanizmów manualnych. W niniejszym rozdziale opisujemy działanie funkcjonalne, każdego z operatorów.

Chcąc ułatwić Tobie podróż przez świat GRAVITY, pogrupowaliśmy operatory kierując się zasadą celu.



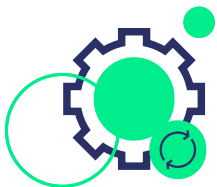
### 7.1 OPERATORY WEJŚCIA

- INPUT DATA
- INPUT FILE
- INPUT MEMORY
- INPUT OTHER PROJECT
- INPUT SQL



### 7.2 OPERATORY WYJŚCIA

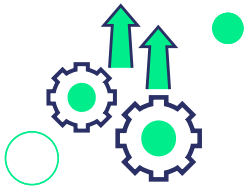
- OUTPUT DATA
- OUTPUT FILE
- OUTPUT MEMORY
- OUTPUT PDF
- OUTPUT POST
- OUTPUT SQL
- OUTPUT TEMPORARY



### 7.3 OPERATORY PRZETWARZANIA

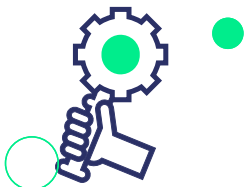
- BUSBAR
- COMPUTING
- DATA CLEAN
- DISCRETISATION
- FILTER
- GROUPING
- IF
- INTEGRATION OF BUSBAR
- MERGE OF BUSBAR
- MULTIPLICATOR

- NORMALIZATION
- OCR
- SORT
- STANDARIZATION
- STANDARIZATION OUTPUT
- STOP
- TEXT DICTIONARY
- TEXT RECOGNIZE
- WIDTH OF BUSBAR
- VALIDATION



## 7.4 OPERATORY PRZETWARZANIA ZEWNĘTRZNEGO

- CALL EXE
- CALL FUNCTION DLL
- CALL SQL
- CALL WEB SERVICE



## 7.5 OPERATORY PROJEKTOWANE WŁASNORĘCZNIE

Operatory mogą być również tworzone przez Ciebie samodzielnie. Utworzone ręcznie operatory, są obecne w systemie na takich samych prawach jak operatory wbudowane. Tworzenie operatora polega na zadeklarowaniu nazwy oraz kodu przetwarzania w środowisku GRAVITY. Podczas kodowania masz dostęp do strumienia danych na magistrali wejściowej oraz wskazujesz sposób przetwarzania strumienia wyjściowego. Kod jest automatycznie kompilowany w trakcie wywołania aplikacji GRAVITY.

## 8.

## OPIS OPERATORÓW GRAVITY

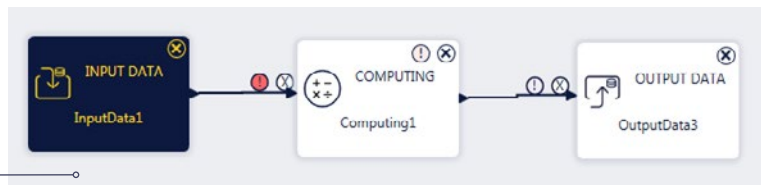
## 8.1 OPERATORY WEJŚCIA

## 8.1.1 INPUT DATA

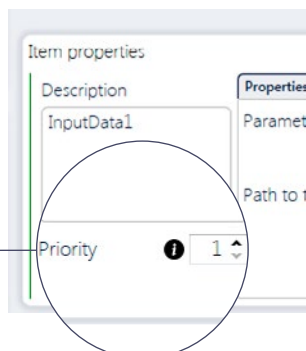
**i** W kontekście wywołania aplikacji GRAVITY jako funkcji WEB SERVICE przez zewnętrzne oprogramowanie masz do dyspozycji dwa operatory komunikujące się z wnętrzem projektu: INPUT DATA oraz OUTPUT DATA.

INPUT DATA to operator pobierający strumień danych z programu wywołującego projekt GRAVITY, jako funkcję WEB SERVICE, przekazanych do projektu w protokole JSON. INPUT DATA może również ustawiać wartości parametrów globalnych obecnych w aplikacji GRAVITY.

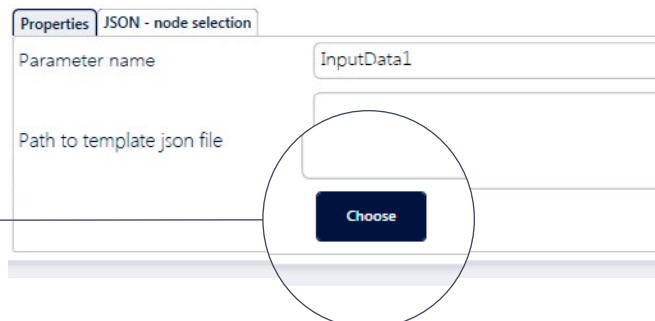
Na podstawie pliku wzorcowego, GRAVITY stworzy listę potencjalnych elementów formatu JSON. Zaznaczając, deklarujesz dokładną strukturę danych przekazywanych do GRAVITY przez system wywołujący aplikację GRAVITY jako funkcję WEB SERVICE.



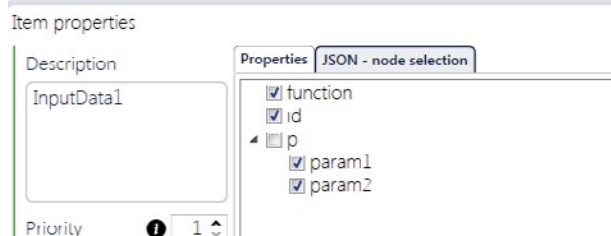
Przykład projektu GRAVITY z użyciem operatora INPUT DATA.



Ponieważ operator INPUT DATA jest operatorem początkowym, a w projekcie może istnieć wiele operatorów początkowych, możesz zadeklarować kolejność wykonania operatorów początkowych poprzez edycję parametru PRIORITY.



W pierwszym kroku wskaż przykładowy plik o formacie JSON. Wybrany plik będzie wzorcem struktury danych wejściowych.





Na podstawie pliku wzorcowego ACTIVIO stworzy listę potencjalnych elementów formatu **JSON**. Zaznaczając deklarujesz dokładną strukturę danych przekazywanych do GRAVITY przez system wywołujący aplikację GRAVITY jako funkcję **WEB SERVICE**.

Poniżej format JSON danych przekazywanych podczas wywołania **WEB SERVICE** przez program zewnętrzny:

- `call_alias` – nazwa **WEB SERVICE** (patrz opcja **CALLS**, `execution_type = Webservice`)
- `process_id` – nieobligatoryjna zmienna sesyjna, nadawana opcjonalnie przez program wywołujący, wykorzystywana podczas stronicowania (paczkowania danych);
- `get_data` – lista kontroltek **OUTPUT DATA** do pobrania danych (w jednym projekcie może być wiele operatorów **OUTPUT DATA**)
  - » `records_count` – ilość rekordów w paczce (stronie); brak parametru oznacza rezygnację z paczkowania;
  - » `description` – nazwa kontrolki **OUTPUT DATA**
  - » `offset` – od którego rekordu system ma rozpocząć stronicowanie
- `parameters`
  - » parametry wejściowe globalne; jeżeli wcześniej zdefiniujesz parametry globalne dla aplikacji GRAVITY, to wskazując ich unikalną nazwę w pliku **JSON** wraz z wartością przeprowadzasz akcję podstawienie przekazanej wartości do parametrów globalnych (te parametry globalne mogą być użyte np. w innym operatorze np. w **INPUT SQL**);
  - » parametry wejściowe dla wyspecyfikowanej kontrolki **INPUT DATA** (identyfikacja operatora **INPUT DATA** według nazwy): na magistrali wyjściowej wskazanej z nazwy kontrolki **INPUT DATA** znajdzie się cały strumień danych przekazanych w ten sposób

Przykładowy format JSON (dane przekazywane podczas wywołania **WEB SERVICE**):

```
{
  „call_alias”: „test_index”,
  „process_id”: „327”,
  „get_data”: [
    {
      «description»: «OutputData1»,
      „records_count”: „10”,
      „offset”: „10”
    }
  ],
  „parameters”: {
    „id”: „4”,
    „id2”: „1000”,
    «input_data_1»: [
      {
        „function”: „z22”,
        „id”: „84576”
      },
      {
        „function”: „p23”,
        „id”: „31”
      }
    ]
  }
}
```

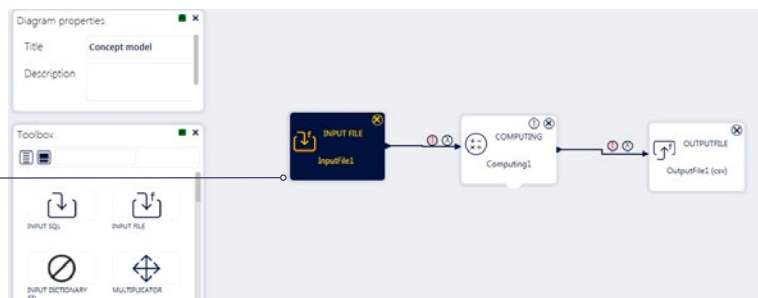
### 8.1.2 INPUT FILE



Operator INPUT FILE umożliwia pobranie danych z pliku. Na wejściu operatora jest wybrany plik danych, na wyjściu znajduje się magistrala ze strumieniem danych.

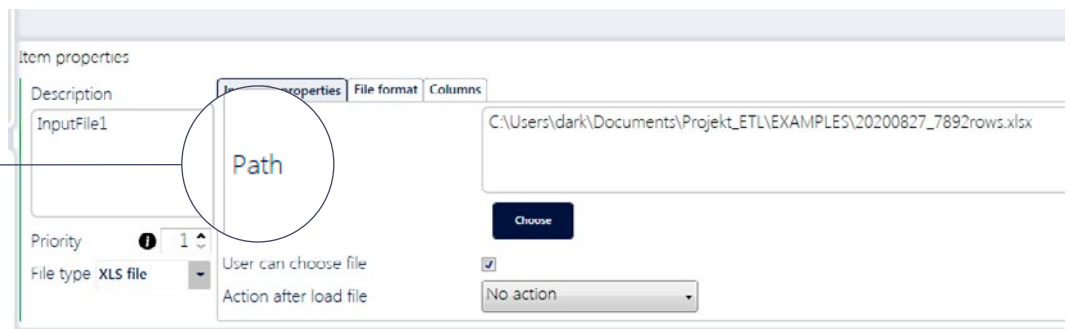
#### USTAWIENIE PARAMETRÓW PLIKU WEJŚCIOWEGO

Przykład projektu GRAVITY z użyciem operatora INPUT FILE



W zakładce **INPUT FILE PROPERTIES**, ustawisz niezbędne parametry pozwalające zidentyfikować plik pobierany przez operator.

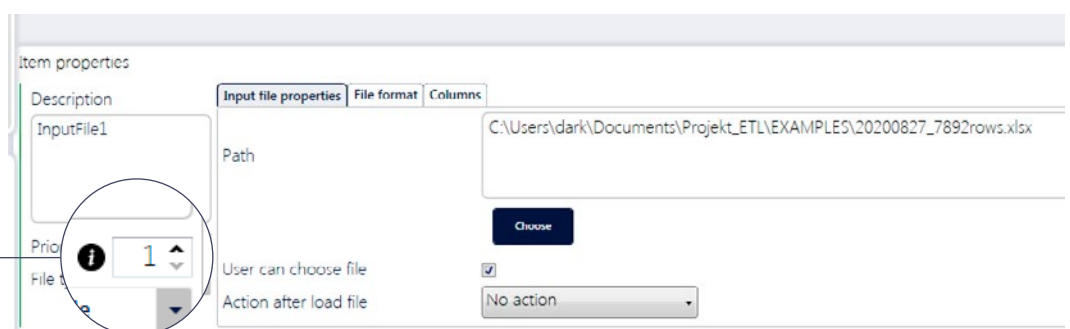
Na ilustracji zaznaczyliśmy parametr PATH: to tutaj wskazujesz lokalizację pliku.



Wybór pliku może być dokonywany w bardziej wyrafinowany sposób. Opcja **FILE RECOGNITION**, pozwala na następujące warianty postępowania:

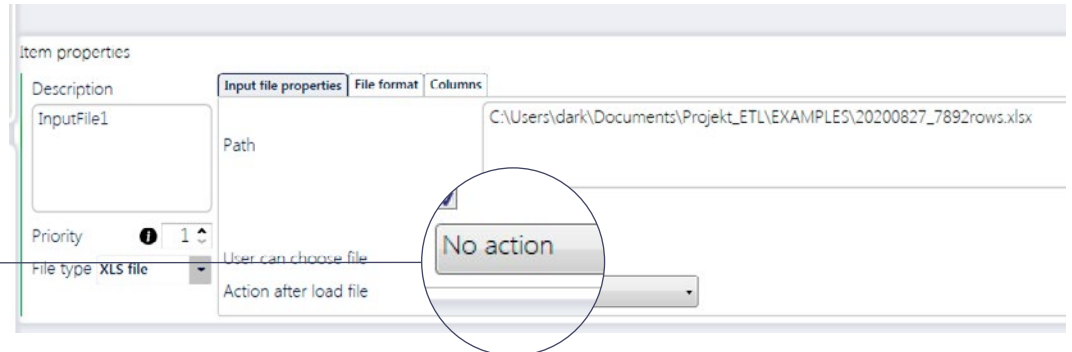
- **ONLY PATTERN** .....GRAVITY pobiera wówczas plik jednoznacznie wskazany w parametrze PATH
- **USER CAN CHOOSE**....użytkownik sam wskaże plik podczas akcji (akcja ta ma sens tylko wówczas, gdy wywołujemy projekt manualnie)
- **OLDEST IN CHOSEN PATH**.....GRAVITY automatycznie wybiera najstarszy plik z lokalizacji (PATH)

Ponieważ operator INPUT FILE jest operatorem początkowym, a w projekcie może istnieć wiele operatorów początkowych możesz zadeklarować kolejność wykonania operatorów początkowych poprzez edycję parametru PRIORITY



Przetwarzanie pobranego pliku może wymagać różnych akcji, po pobraniu pliku do przetwarzania.

GRAVITY daje Tobie następujące możliwości

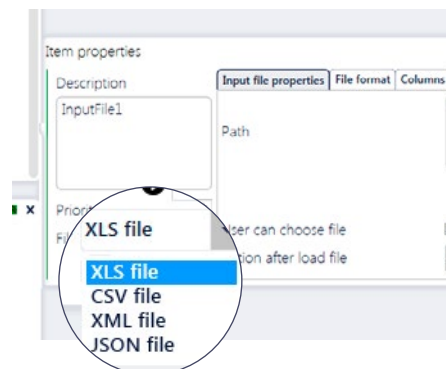


- **NO ACTION** ... plik zostaje pobrane i pozostaje w swojej lokalizacji
- **CHANGE FILE NAME**....plik pozostaje w swojej lokalizacji, jednak jego nazwa jest zmieniana

Uwaga: nową nazwę pliku można zaprojektować używając zmiennych (zakres zmiennych dostępny w pomocy kontekstowej)

- **MOVE FILE**.....plik zostaje przeniesiony do innej lokalizacji
- **REMOVE FILE**.....plik zostaje usunięty po pobraniu

GRAVITY pozwala na pobieranie plików w następujących formatach:



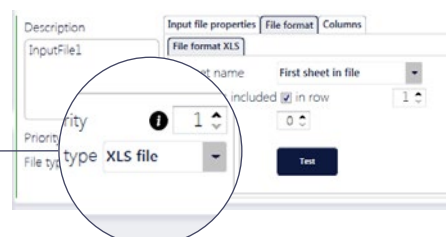
- **XLS**
- **CSV**
- **XML**
- **JSON**

### KONFIGURACJA FORMATU PLIKU WEJŚCIOWEGO

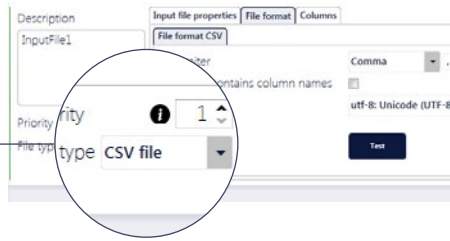
W zależności od formatu pliku pobieranego, użytkownik jest zobowiązany do ustawienia różnych parametrów.

### DEKLARACJA KOLUMN PLIKU WEJŚCIOWEGO

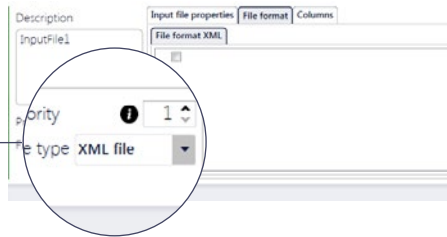
Na ilustracji konfiguracja formatu pliku XLS.



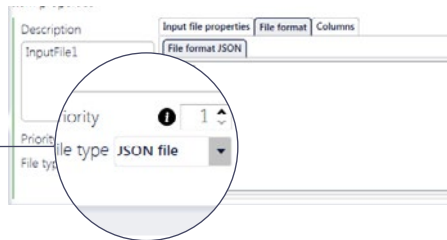
Na ilustracji konfiguracja formatu pliku CSV



Konfiguracja formatu pliku XML (GRAVITY daje możliwość użycia specjalnego edytora do konfiguracji pliku XML).

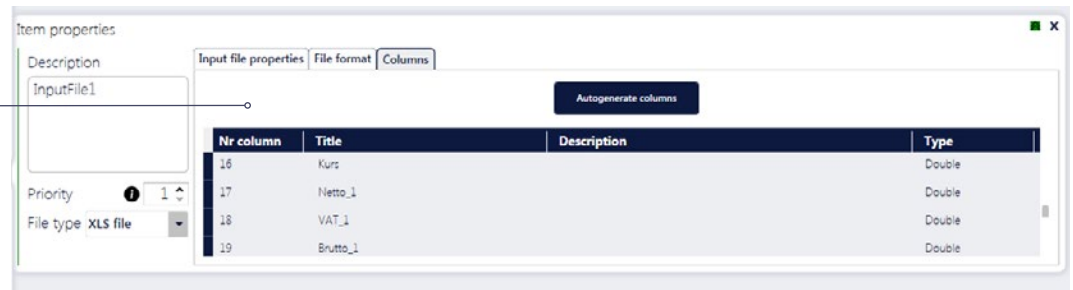


Konfiguracja formatu pliku JSON (GRAVITY daje możliwość użycia specjalnego edytora do konfiguracji pliku JSON).



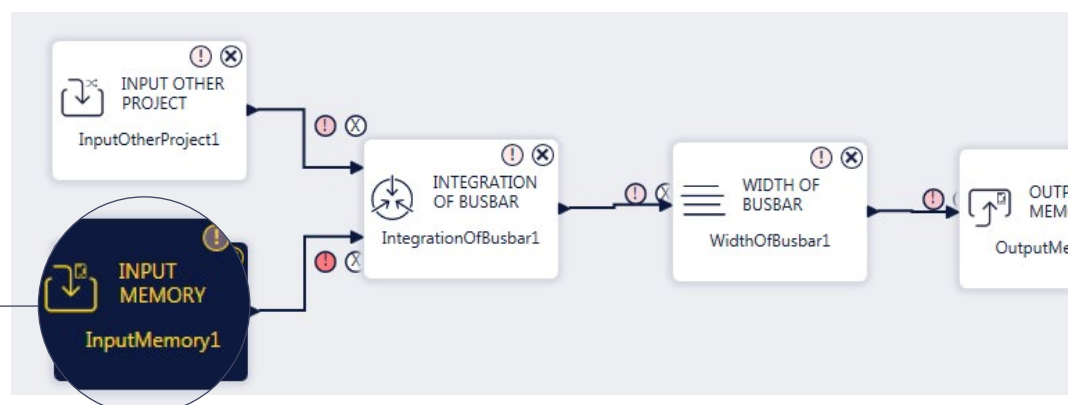
### 8.1.3 INPUT MEMORY

Jeżeli operator jest już podłączony do magistrali, będziesz mógł rozpoznać kolumny automatycznie.

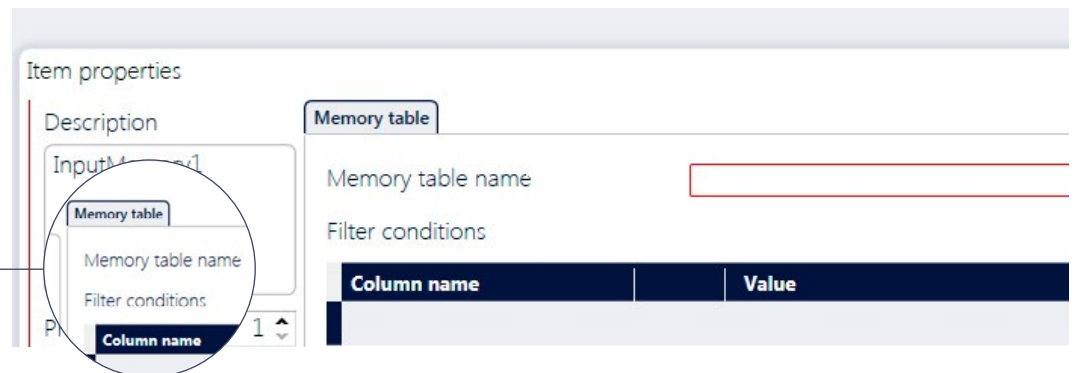


Operator INPUT MEMORY wprowadza nowy strumień danych na magistralę wyjściową. Strumień danych pochodzi ze zbioru pamięciowego, zbudowanego przez operator OUTPUT MEMORY, w tej samej lub innej aplikacji GRAVITY.

Przykład projektu GRAVITY z użyciem operatora INPUT MEMORY.

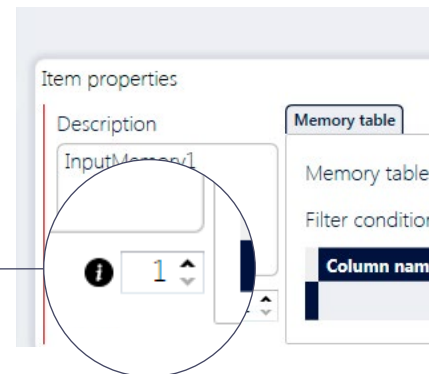


## KONFIGURACJA OPERATORA



Musisz wskazać unikalną nazwę zbioru pamięciowego.

Zwróć uwagę, że nie musisz zawsze pobierać wszystkich danych zgromadzonych w zbiorze pamięciowym, lecz możesz pobrać ich część, używając warunków filtrujących.



Możesz zadeklarować kolejność wykonania operatorów początkowych poprzez edycję parametru PRIORITY

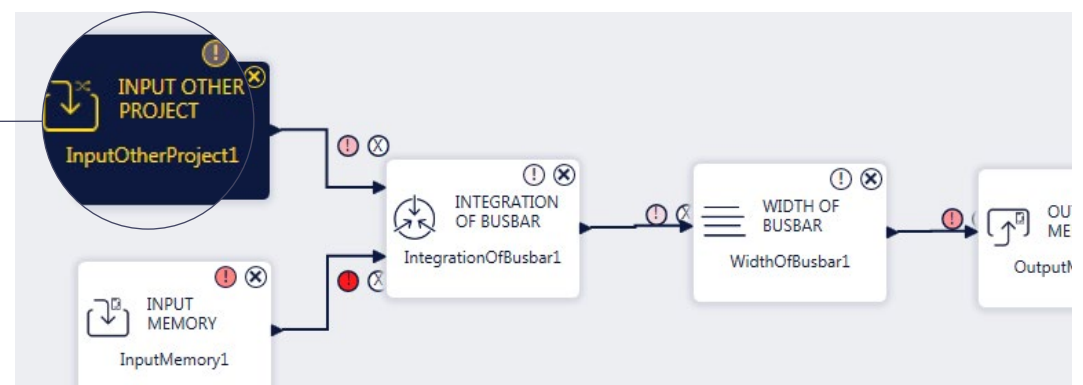
Ponieważ operator **INPUT MEMORY** jest operatorem początkowym, a w projekcie może istnieć wiele operatorów początkowych.

### 8.1.4 INPUT OTHER PROJECT



Operator **INPUT OTHER PROJECT** umożliwia wprowadzenie do magistrali danych, strumienia danych będącego wynikiem przetwarzania innego projektu GRAVITY.

## KONFIGURACJA OPERATORA



Przykład projektu GRAVITY z użyciem operatora **INPUT OTHER PROJECT**.

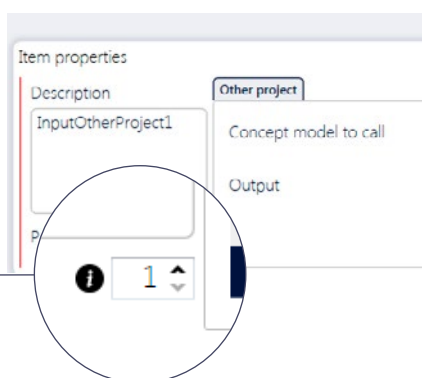
W operatorze **INPUT OTHER PROJECT** musisz zadeklarować inny projekt GRAVITY. Projekt ten zostanie

Musisz wskazać operator typu OUTPUT, czyli źródło pochodzenia danych z zewnętrznego projektu.



automatycznie przetworzony w trakcie realizacji aktualnego projektu, w celu uzyskania strumienia danych początkowych.

Możesz zadeklarować kolejność wykonania operatorów początkowych poprzez edycję parametru PRIORITY



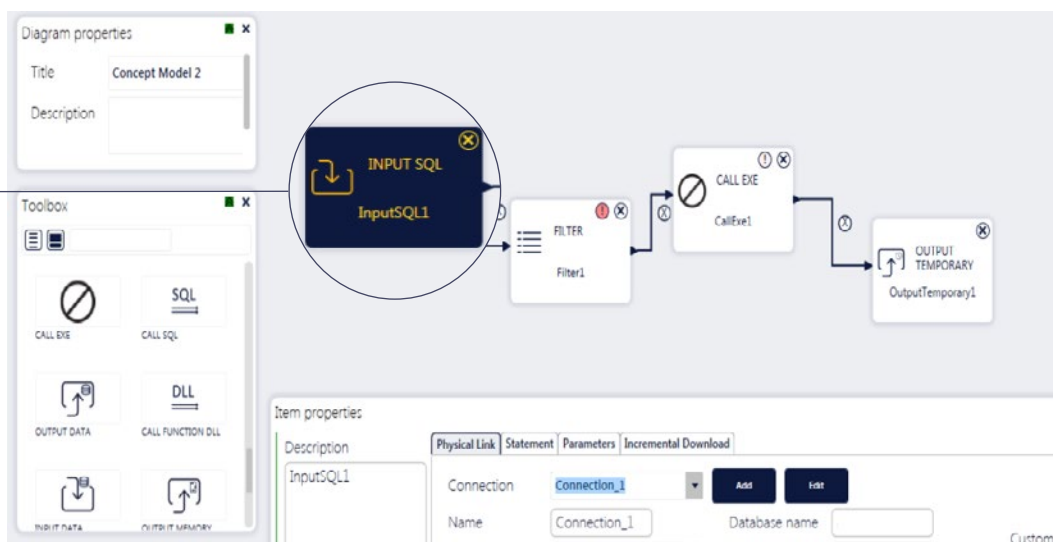
Ponieważ operator **INPUT OTHER PROJECT** jest operatorem początkowym, a w projekcie może istnieć wiele operatorów początkowych.

### 8.1.5 INPUT SQL

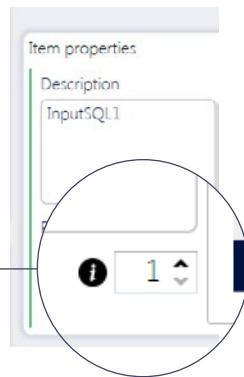


Operator **INPUT SQL** umożliwia pobranie danych z zewnętrznych źródeł danych (bazy SQL).

Projekt GRAVITY z użyciem operatora **INPUT SQL**

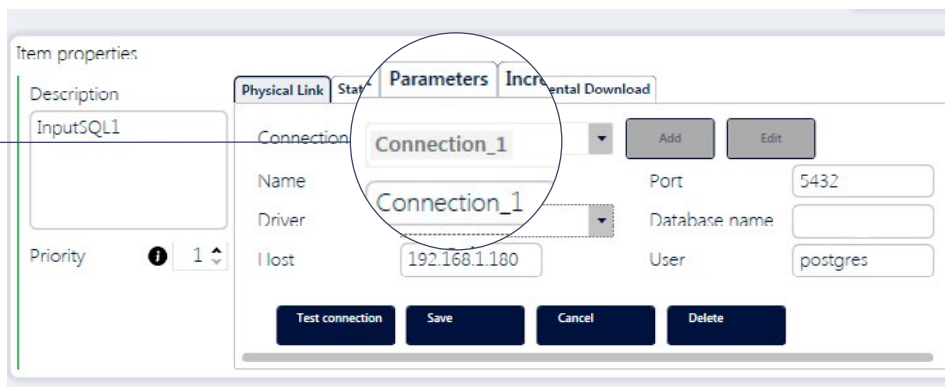


### USTAWIENIE PARAMETRU: PRIORITY

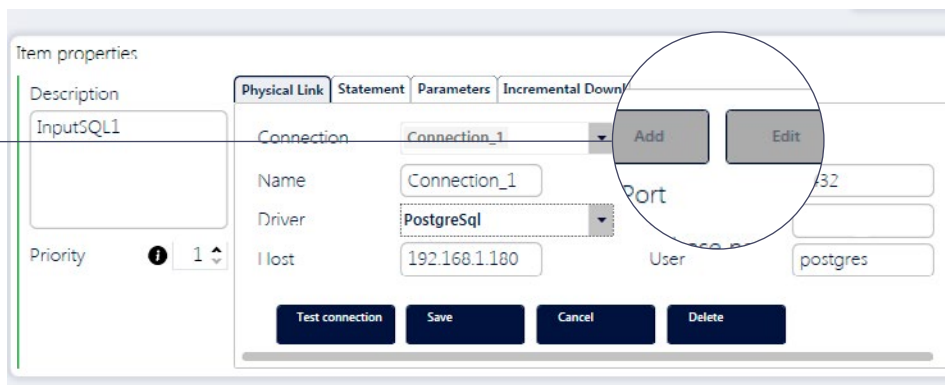


Jeżeli w projekcie GRAVITY jest wiele operatorów wejściowych, możesz ustawić kolejność przetwarzania ustawiając parametr PRIORITY.

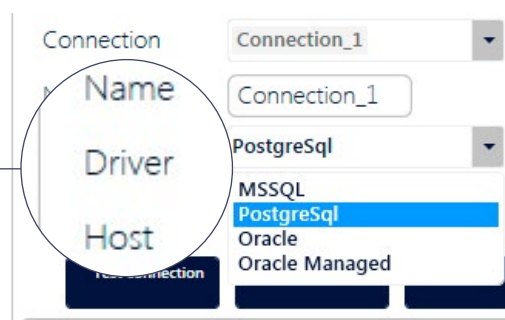
### USTAWIENIE GRUPY PARAMETRÓW: PHYSICAL LINK



Jeżeli wcześniej kiedykolwiek skonfigurowano połączenie z bazą SQL, można je wskazać poprzez wybór ze słownika.



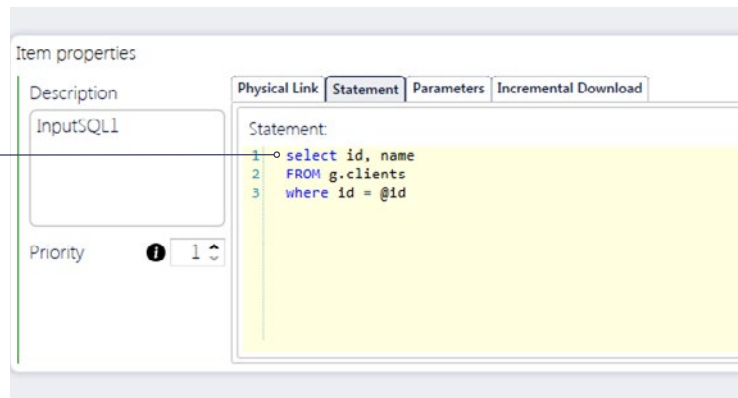
W przypadku, gdy chcesz skonfigurować nowe połączenie lub edytować stare musisz użyć opcji (ADD lub EDIT).



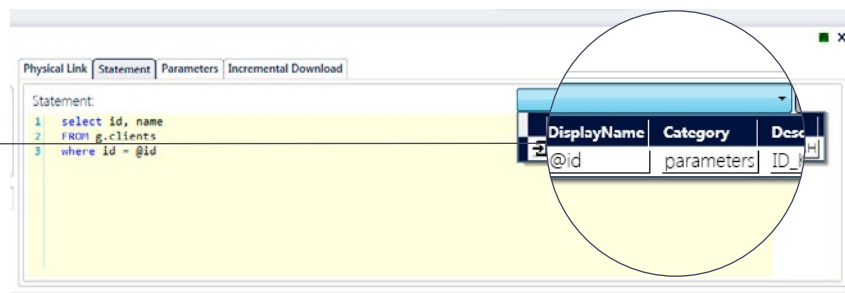
Podczas edycji parametrów połączenia, możesz wybrać sterowniki ze słownika

### USTAWIENIE PARAMETRU: STATMENT

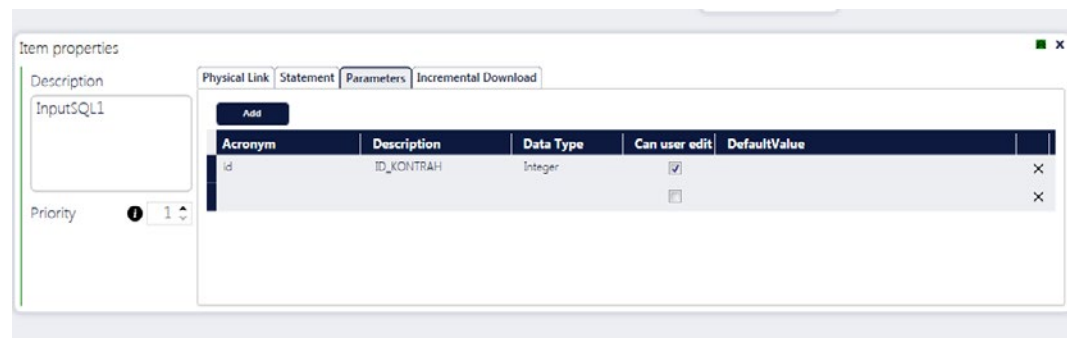
Edycja wyrażenia SELECT.



W wyrażeniu SELECT możesz użyć parametrów zdefiniowanych w zakładce PARAMETERS. Wybór parametru, możesz dokonać poprzez wskazanie słownika



### USTAWIENIE GRUPY PARAMETRÓW: PARAMETERS

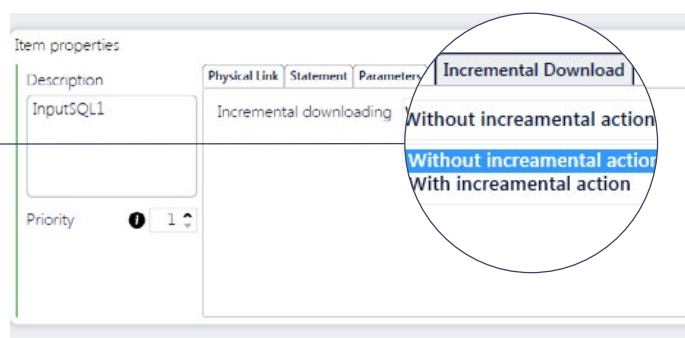


Acronym	Description	Data Type	Can user edit	DefaultValue
id	ID_KONTRAH	Integer	<input checked="" type="checkbox"/>	

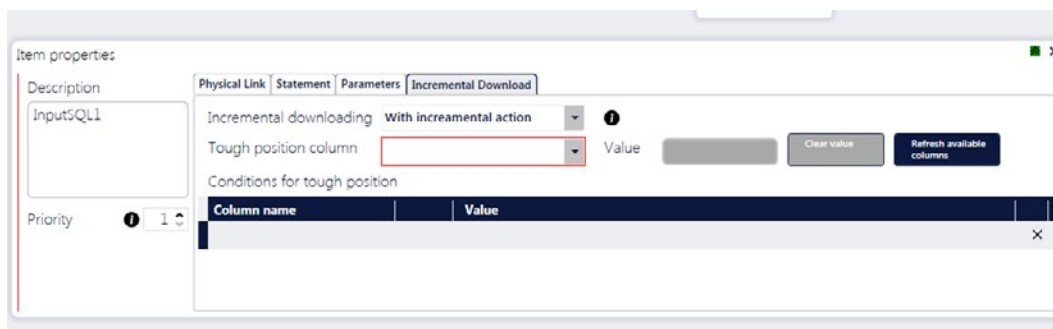
Możesz zadeklarować dowolną liczbę parametrów, które mogą (lecz nie muszą) być edytowane w trakcie przetwarzania przez użytkownika oraz są dostępne dla Twojego wyrażenia SQL, dla każdego operatora projektu.

### USTAWIENIE GRUPY PARAMETRÓW: INCREMENTAL DOWNLOAD

Pobieranie danych może odbywać się drogą przyrostową: wystarczy ustawić na podstawie dostępnego słownika parametr INCREMENTAL DOWNLOADING = WITH INCREMENTAL ACTION







Item properties

Description: InputSQL

Priority: 1

Physical Link | Statement | Parameters | Incremental Download

Incremental downloading: With incremental action

Tough position column: [Empty dropdown] Value: [Input field] [Clear value] [Refresh available columns]

Conditions for tough position

Column name	Value
-------------	-------

W sytuacji, kiedy zdecydujesz się na metodę przyrostową, musisz wskazać kolumnę kluczową (tough position column). Dodatkowo, możesz wskazać warunki, które muszą być spełnione, aby system przyjął wartość kolumny jako ostatnią, pobraną po zakończeniu procesu przetwarzania. Wartość ostatnio pobrana, może być czyszczona za pomocą opcji **CLEAR VALUE**.



Deklaracja metody przyrostowej nie jest widoczna w opcji STATEMENT w postaci kodu SQL. Kod SQL, odpowiadający za powyższy mechanizm powstaje automatycznie (jest generowany przez GRAVITY) podczas akcji pobierania danych, zawężając pobierany strumień danych.



## 8.2 OPERATORY WYJŚCIA

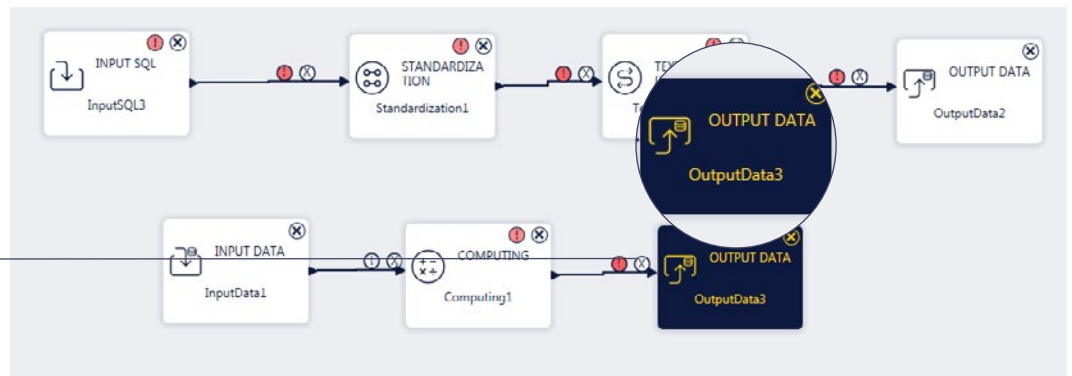
### 8.2.1 OUTPUT DATA



W kontekście wywołania aplikacji GRAVITY jako funkcji WEB SERVICE przez zewnętrzne oprogramowanie masz do dyspozycji dwa operatora komunikujące się z wnętrzem projektu: INPUT DATA oraz OUTPUT DATA.

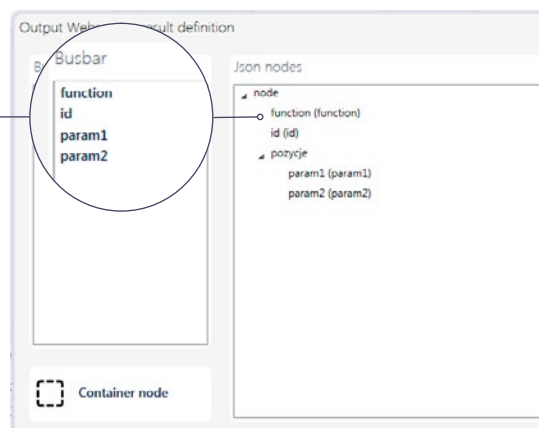
OUTPUT DATA to operator, przekazujący strumień danych, jako rezultat przetwarzania aplikacji GRAVITY, do programu wywołującego projekt, jako funkcję WEB SERVICE. Strumień danych jest w formacie JSON.

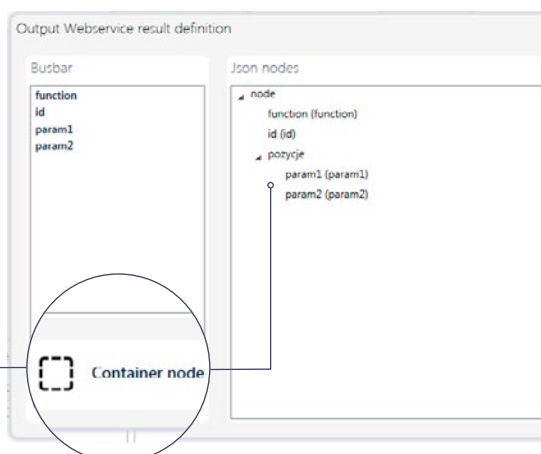
Przykład projektu GRAVITY z użyciem operatora OUTPUT DATA.



W pierwszym kroku musisz wybrać opcję DEFINNE JSON, która pomoże Tobie w automatyczny i przejrzysty sposób skonfigurować format strumienia danych wyjściowych.

Format budujesz poprzez akcję drag and drop kolumn sygnału wejściowego na obszar formatu JSON





Możesz tworzyć również zagłębienia poprzez akcję drag and drop obszaru CONTAINER NODE, na obszar formatu JSON.

Poniżej format JSON danych przekazywanych jako rezultat, podczas wywołania WEB SERVICE przez program zewnętrzny:

- error\_code – numer błędu (0 – brak błędu)
- exeptions – tablica błędów (tylko na użytek klasy EXEPTION C#);
- data – tablica danych z wszystkich operatorów OUTPUT DATA zadeklarowanych podczas wywołania funkcji WEB SERVICE (patrz operator INPUT DATA)
  - » [nazwa operatora]
  - » (tablica danych ze wskazanego operatora OUTPUT DATA)

Przykładowy format JSON (strumień danych jako rezultat):

```

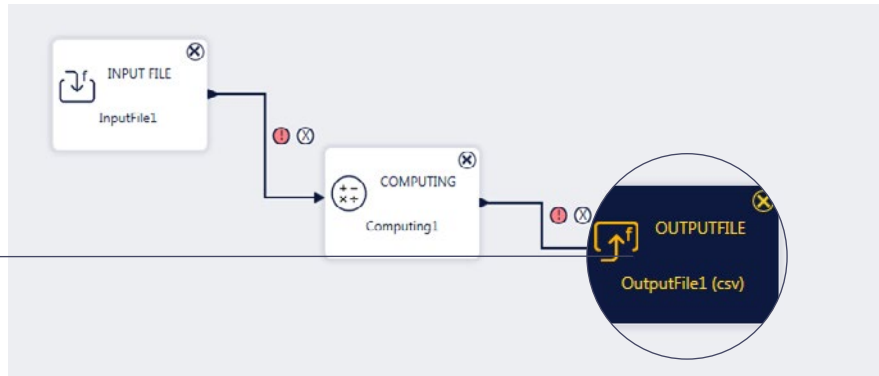
"ErrorCode": 0,
"Exceptions": [],
"Data": [
  {"OutputDataIndeksy": {
    "data": [
      {"id_index": "4",
        "index": "Product4",
        "name": "Name_ Product 4",
        "jm": "kg"
      },
      {
        "id_index": "6",
        "index": " Product 6",
        "name": "name_ Product6 ",
        "jm": "kg"
      }
    ]
  }
},
  {
    "OutputData1": {
      "node": [
        {
          "index": "Product5 ",
          "name": " name_Product 5"
        },
        {
          "index": " Product10 ",
          "name": "name_ Product10 "
        },
        {
          "index": " Product 11",
          "name": " name_Product11 "}
      ]
    }
  }
]
}

```

## 8.2.2 OUTPUT FILE

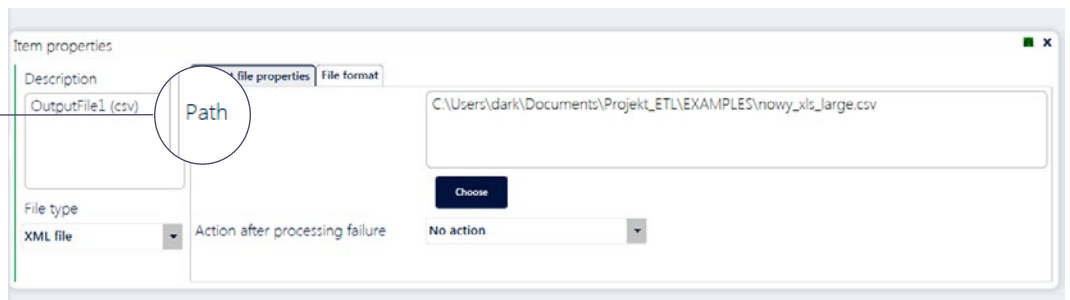


Operator OUTPUT FILE umożliwia utworzenie pliku z danymi. Na wejściu operatora jest strumień danych, który w wyniku działania operatora zostaje przekształcony na plik, w formacie wskazanym przez użytkownika.

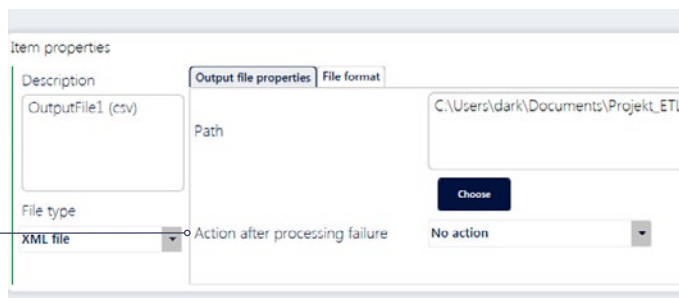


Przykład projektu GRAVITY z użyciem operatora OUTPUT FILE.

### USTAWIENIE PARAMETRÓW PLIKU WYJŚCIOWEGO



parametr PATH: to tutaj wskazujesz lokalizację oraz nazwę pliku.



W opcji wskazanej, zamodelujesz wariant akcji, w przypadku niepowodzenia przetwarzania.

W zakładce OUTPUT FILE PROPERTIES ustawisz niezbędne parametry pliku generowanego przez operator.

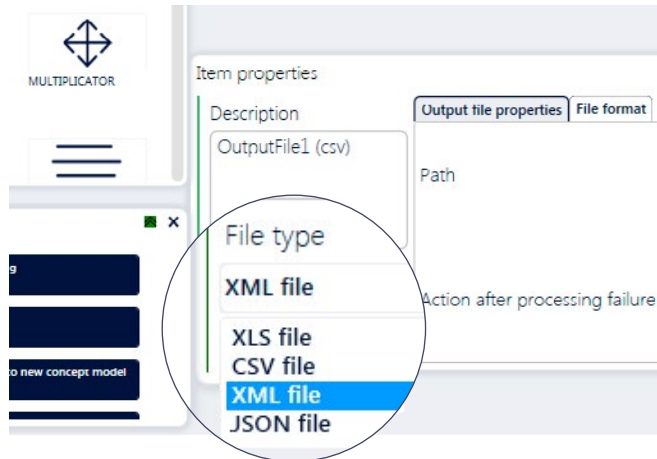
Standardowo, przetwarzanie tworzy nowy plik lub go nadpisuje, jeżeli znajdzie identyczny plik w dotychczasowej lokalizacji.

Możesz zadeklarować akcje:

- **NO ACTION** przetwarzanie nie generuje nowego pliku
- **CREATE OR OVERWRITE EMPTY FILE** wygenerowany plik będzie pusty
- **REMOVE OLD FILE** jeżeli istnieje plik we wskazanej lokalizacji o tej samej nazwie, zostanie wykasowany

GRAVITY pozwala na generowanie plików w następujących formatach:

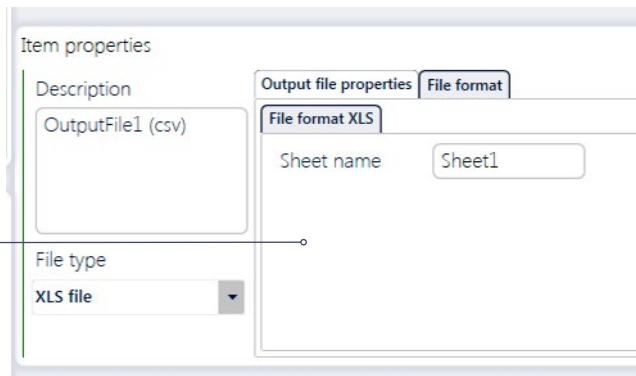
- XLS
- CSV
- XML
- JSON



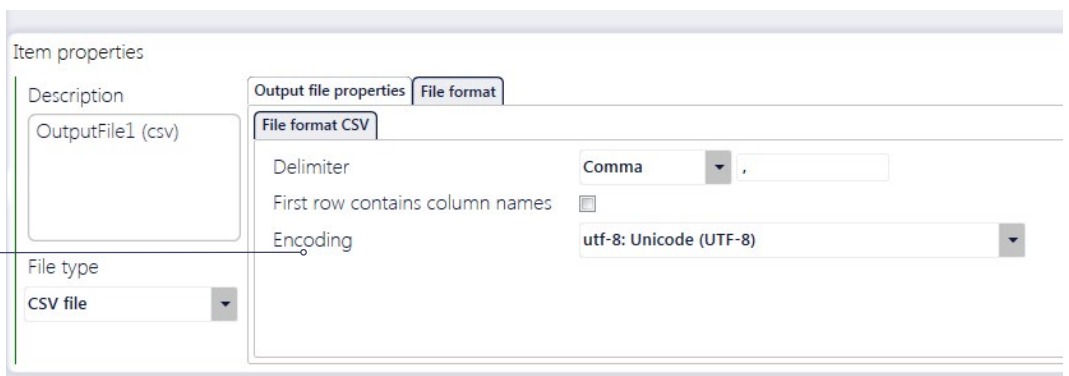
### KONFIGURACJA FORMATU PLIKU WYJŚCIOWEGO

W zależności od formatu pliku wyjściowego użytkownik zobowiązany jest do ustawienia różnych parametrów.

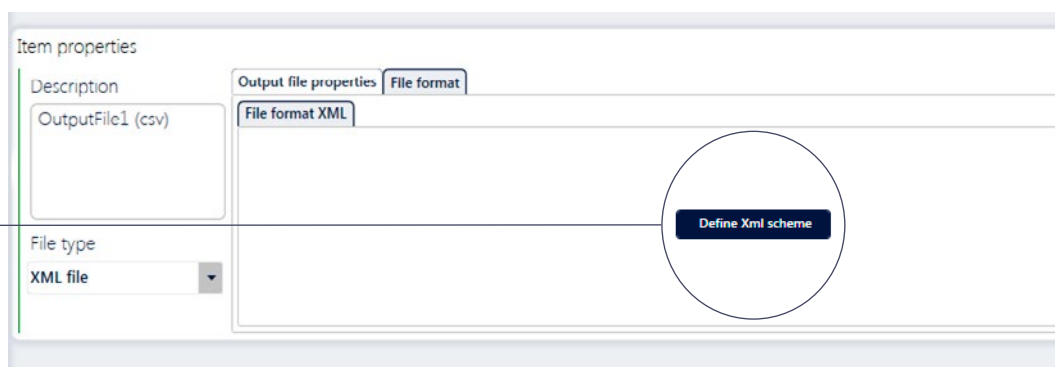
Konfiguracja formatu pliku XML..



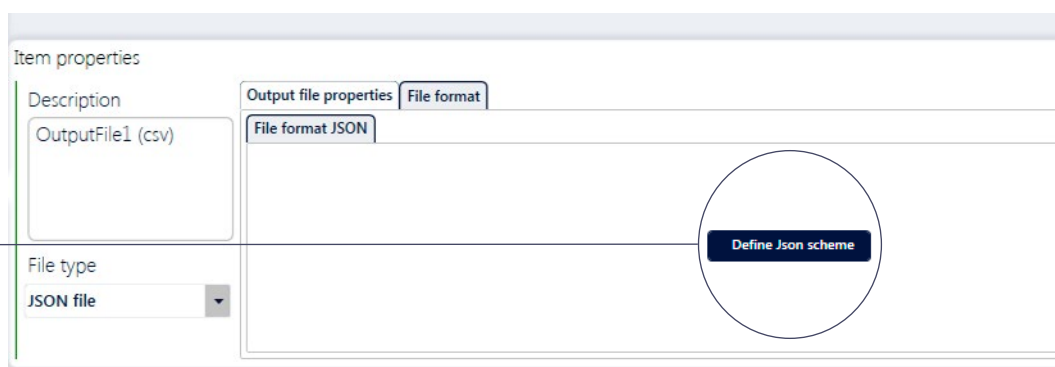
Konfiguracja formatu pliku CSV



Konfiguracja formatu pliku XML (GRAVITY daje możliwość użycia specjalnego edytora do konfiguracji pliku XML).



Konfiguracja formatu pliku JSON (GRAVITY daje możliwość użycia specjalnego edytora do konfiguracji pliku JSON).

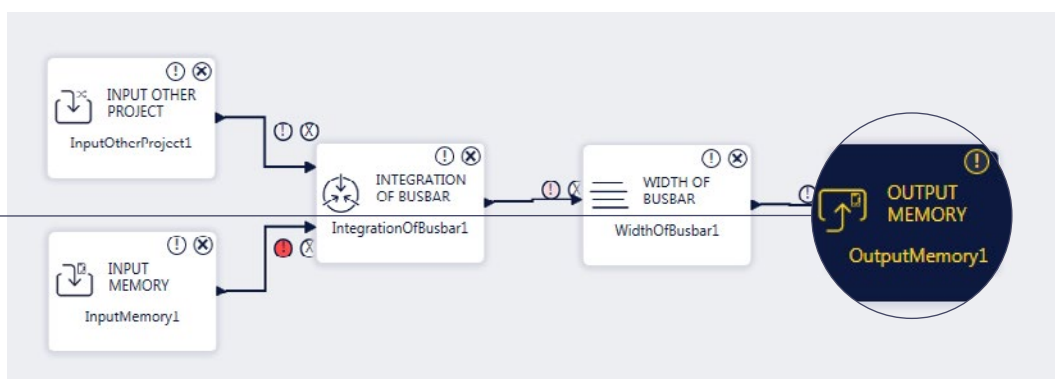


### 8.2.3 OUTPUT MEMORY



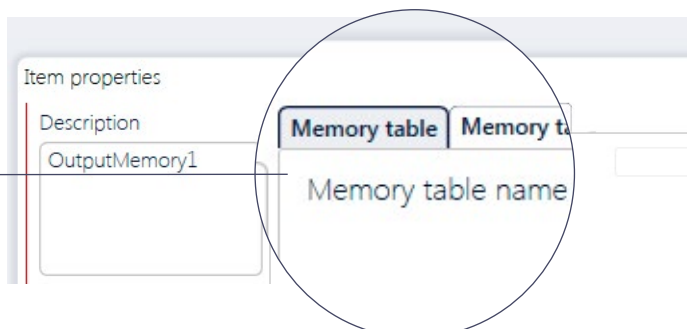
Operator OUTPUT MEMORY gromadzi dane, pochodzące ze strumienia danych magistrali wejściowej. Zgromadzone dane mogą być wykorzystywane w tej samej aplikacji GRAVITY lub w innych aplikacjach GRAVITY.

Przykład projektu GRAVITY z użyciem operatora OUTPUT MEMORY.



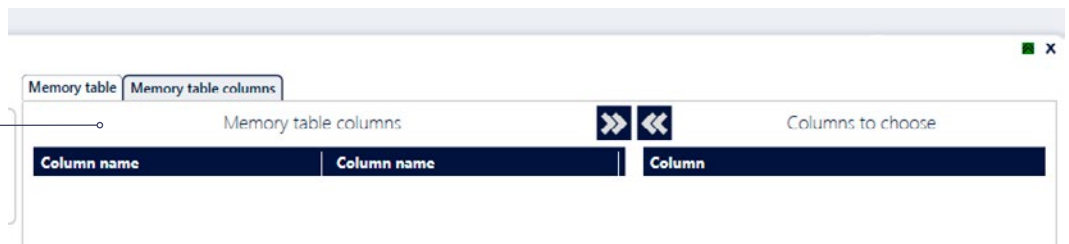
### KONFIGURACJA OPERATORA

Danym zbieranym w pamięć, i musisz nadać jakąś unikalną nazwę.



Do gromadzonych danych będziesz się odwoływał wykorzystując nadaną nazwę.

Zakres gromadzonych danych zależy od konfiguracji. Możesz wskazać dowolną liczbę kolumn z magistrali wchodzącej do operatora (akcja drag and drop).



#### 8.2.4 OUTPUT PDF



Operator OUTPUT PDF umożliwia sformatowanie danych z magistrali wyjściowej w taki sposób, że rezultatem działania operatora będzie jeden obiekt w formie dokumentu PDF. Dokument PDF może zawierać dowolne dane znajdujące się na magistrali wejściowej. Forma dokumentu PDF jest konfigurowana podczas tworzenia aplikacji GRAVITY.

Operator umożliwia stworzenie w GRAVITY kompletnego systemu raportowania, wydruków, zestawień, w szacie dostosowanej do potrzeb użytkownika. Raporty są dostępne dla każdego oprogramowania zewnętrznego jako funkcje [WEB SERVICE](#).

Szczegółowy opis budowy raportu (a więc konfiguracji kontrolki [OUTPUT PDF](#)) znajduje się w niezależnym podręczniku [GRAVITY REPORT](#).

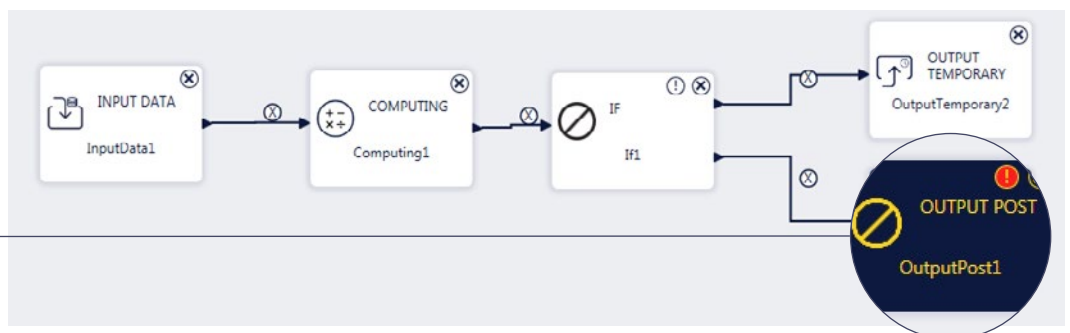
#### 8.2.5 OUTPUT POST



Operator OUTPUT POST daje możliwość przetworzenia sygnału na magistrali wejściowej, na informację wysłaną pocztą elektroniczną.

Adresaci poczty mogą być ustaleny na podstawie informacji zawartej w magistrali wejściowej lub określony w sposób trwały. Dane strumienia wejściowego mogą być zgromadzone w pliku i wysłane pocztą jako załącznik.

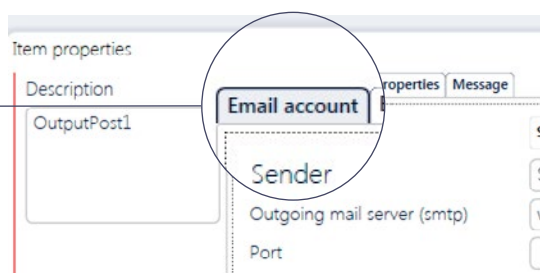
Poczta może być redagowana dla każdego rekordu strumienia danych wejściowych lub grupowana dla zadeklarowanego przez projektanta operatora klucza.



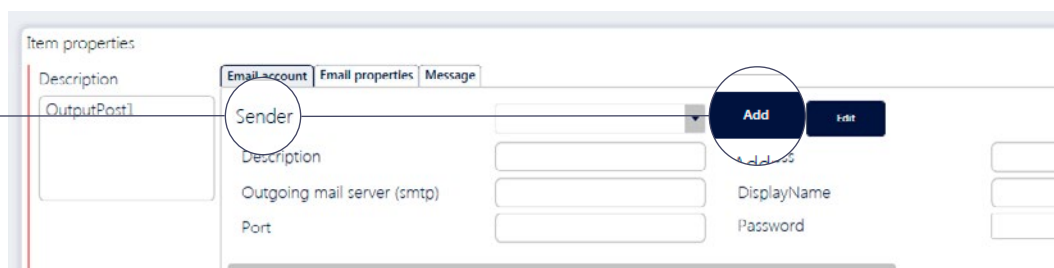
Przykład projektu GRAVITY z użyciem operatora [OUTPUT POST](#).

#### KONFIGURACJA OPERATORA: EMAIL ACCOUNT

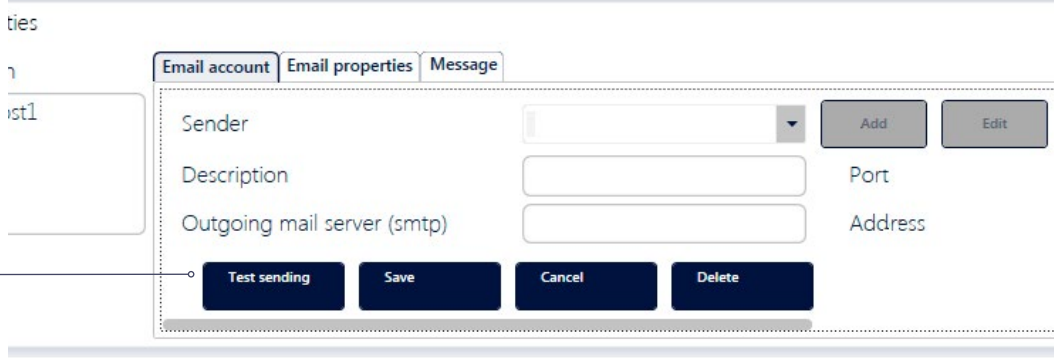
W zakładce [EMAIL ACCOUNT](#) konto skrzynki pocztowej, niezbędne do wysyłania korespondencji mailowej.



Skrzynkę możesz wybrać ze słownika (jeżeli wcześniej konfigurowałeś w GRAVITY skrzynki pocztowe) lub zadeklarować nową skrzynkę



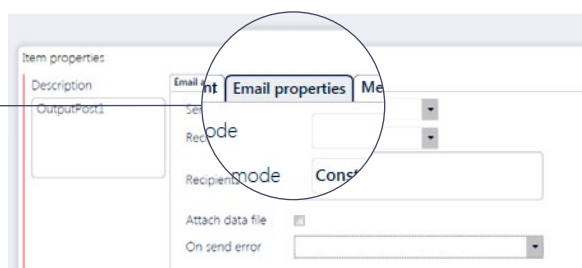
Na ilustracji możesz zobaczyć, jakie parametry należy wypełnić konfigurując skrzynkę pocztową.



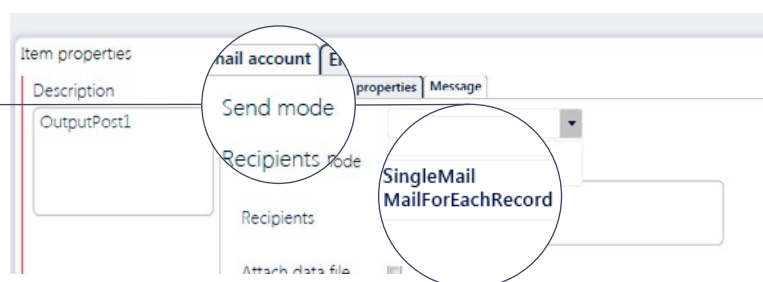
Zauważ, że masz dostępną opcję **TEST SENDING**, która Tobie pozwoli ocenić poprawność wprowadzonych danych.

### KONFIGURACJA OPERATORA: EMAIL PROPERTIES

W zakładce **EMAIL PROPERTIES** zadeklarujesz odbiorców poczty, dołączane pliki oraz zachowanie systemu podczas błędu.



W pierwszym kroku określ, w jaki sposób chcesz wysłać korespondencję mailową



Opcja **SEND MODE**:

- jeden mail dla całego strumienia (**SEND MODE = SingleMail**),
- dla każdego rekordu strumienia danych chcesz wysłać niezależnego maila (**SEND MODE= MailForEachRecord**)

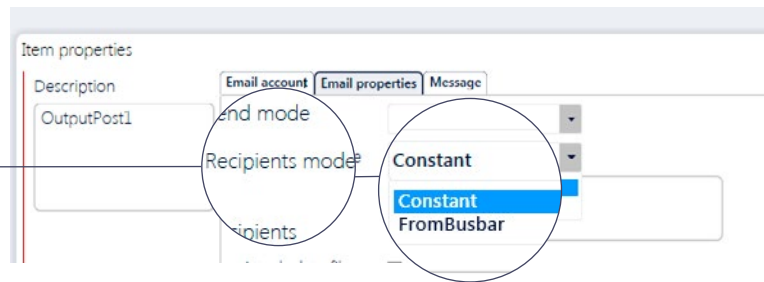
To są dwie kompletnie różne strategie pojmowania procesu z operatorem **OUTPUT POST**, obie drogi postępowania mogą oddać Tobie nieocenione usługi dla niezależnych procesów.



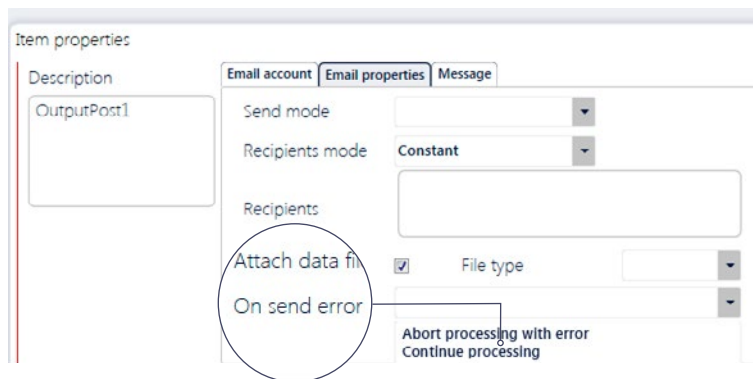
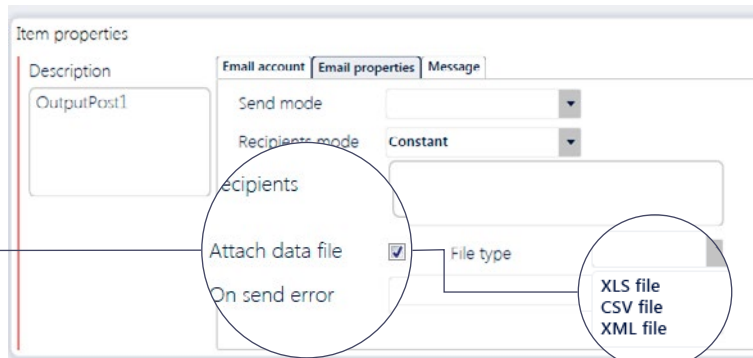
Korzystając z wariantu **SEND MODE= MailForEachRecord** możesz np. generować automatycznie, w wyznaczonych przez Ciebie interwałach czasowych, bez zaangażowania swoich pracowników, salda dla wszystkich swoich kontrahentów, zaś z trybu **SEND MODE = SingleMail** możesz przygotować raport, na podstawie zdiagnozowanej sytuacji, wykrytej automatycznie (alarm wewnętrzny).



Możesz wysyłać pocztę do ustalonych stałych odbiorców (opcja **RECIPIENTS MODE = CONSTANT**;) lub na adres, który znajduje się na szynie danych wejściowych (**RECIPIENTS MODE = FromBusbar**).



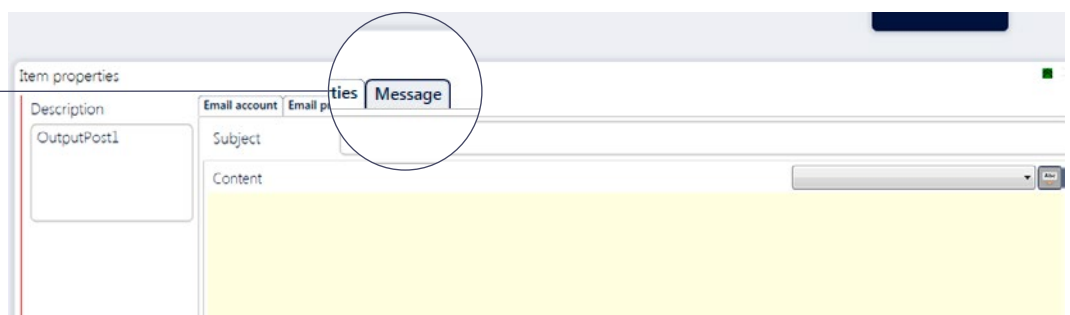
Do korespondencji mailowej możesz dołączyć przetwarzany strumień danych, w wybranym przez Ciebie formacie (patrz ilustracja powyżej). W takim wypadku należy zaznaczyć opcję **ATTACH DATA FILE** oraz wskazać jeden z obsługiwanych przez GRAVITY format pliku: xls, xml lub csv.



Podczas przetwarzania danych w operatorze **OUTPUT POST GRAVITY**, może stanąć przed dylematem w jaki sposób przetwarzać dane w przypadku pojawienia się błędu. Możesz to zaprojektować, ustawiając opcję **ON SEND ERROR** na jeden z dwóch wariantów postępowania:

- **ABORT PROCESSING WITH ERROR**  
(proces jest zastopowany, zaś wszystkie dotychczasowe operacje wycofane – taki proces jest uważany przez nas za niebyły)
- **CONTINUE PROCESSING**

W zakładce **MESSAGE**, zdefiniujesz tytuł oraz treść korespondencji mailowej.





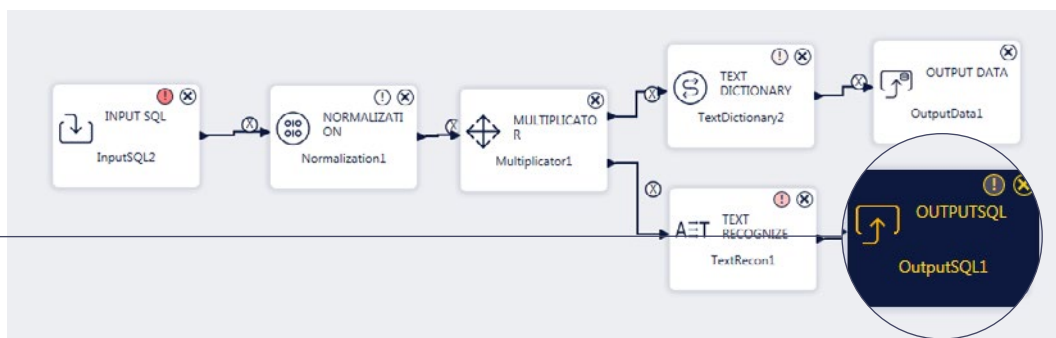
### KONFIGURACJA OPERATORA: MESSAGE

W treści korespondencji możesz użyć parametru **GET\_AGGR** (sumującego wybrany bit na magistrali wejściowej), dzięki czemu możesz odwoływać się do sum w strumieniu danych wejściowych.

### 8.2.6 OUTPUT SQL

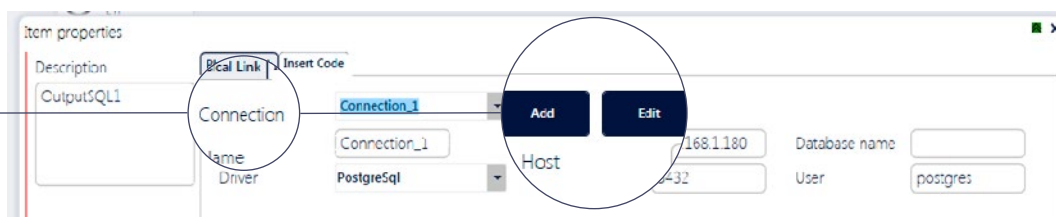


Operator OUTPUT SQL jest odpowiedzialny za wprowadzenie strumienia danych magistrali wejściowej do wybranej bazy SQL.



Przykład projektu GRAVITY z użyciem operatora **OUTPUT SQL**.

### KONFIGURACJA POŁĄCZENIA Z BAZĄ



W pierwszym kroku musisz wskazać połączenie do wybranej bazy. Wyboru dokonujesz ze słownika – jeżeli wcześniej definiowałeś takie połączenia – lub zdefiniować nowe (opcje **ADD** lub **EDIT**).



**UWAGA!** Wskazujesz bazę docelową, do której zamierzasz wprowadzić strumień danych.

### DEKLARACJA AKCJI W PRZYPADKU WYKRYCIA DUPLIKATU



Zdecyduj, co system ma zrobić, w przypadku wykrycia duplikatu danych w bazie docelowej.

Masz do wyboru dwa możliwe zachowania w przypadku wykrycia podczas dodawania danych duplikatów rekordów w bazie docelowej.

#### 📌 Action for the same records

**SKIP** - Skip previously processed records. Do not process them. (If table on lowest level has primary key defined)

**PASS** - Allow all records to processing.

For finding previously processed records we compare the value of primary key (old value - before pk actions executing) with the cache table entry

Jeżeli wybierzesz opcję **PASS**, wówczas zachowanie procesu dla przypadków duplikatów, zależy od akcji wskazanych w opcji **SHOW DIAGRAM TO DEFINE OUTPUT**. W opcji **SQL ACTIONS** możesz zdefiniować:

- **DELETE OLD RECORD** (zastany rekord zostanie skasowany, a nowy rekord dodany) lub
- **UPDATE OLD RECORD** (zastany rekord zostanie nadpisany nowym rekordem)

### DEKLARACJA KODU DODAWANIA STRUMIENIA DANYCH

Wygenerowany automatycznie kod ma formę **XML** i może być przez ciebie modyfikowany również ręcznie.



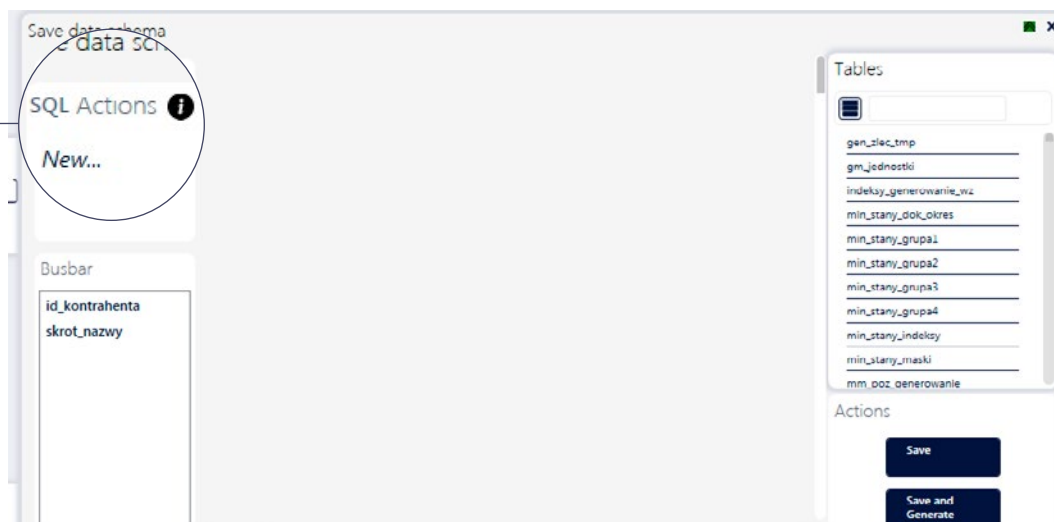
Kod dodania strumienia danych powstaje automatycznie, na podstawie skonfigurowanego przez Ciebie diagramu (patrz opcja **SHOW DIAGRAM TO DEFINE OUTPUT**).

Jeżeli chcesz wrócić do kodu szablonowego powstałego na podstawie diagramu, wystarczy uruchomić opcję **REFRESH INSERT CODE**.

### DEKLARACJA KODU DODAWANIA STRUMIENIA DANYCH: AKCJE DODATKOWE

W pierwszym kroku musisz się zastanowić, jakie akcje chcesz przeprowadzić, przed procesem dodania

Po wybraniu opcji **SHOW DIAGRAM TO DEFINE OUTPUT**, zobaczysz okno jak na ilustracji.



strumienia danych do bazy. Na ilustracji powyżej widzisz zaznaczoną opcję **NEW**, która pozwoli Tobie dodać akcje:

Typu **PRE**, czyli akcje przed dodaniem strumienia danych do bazy;

- Typu **POST**, czyli akcje po dodaniu strumienia danych do bazy;
- Typu **IN**, akcje w trakcie dodawania np. zachowanie systemu w przypadku rozpoznania identycznego rekordu (według **PRIMARY KEY**) w bazie docelowej.

Poniżej objaśnimy szczegółowo działanie każdej akcji.

### **i** Primary key actions

Actions executed on the target or source database.

Depending on the type, can be executed before/after/or as part of the main table modification query (INSERT)

<i>Delete all records [PRE]</i>	- remove all records from target tables
<i>Delete declared space [PRE]</i>	- remove record from declared range - between given pk values
<i>Delete space min max pk [PRE]</i>	- take minimum and maximum pk values from in-data and remove from table records between these values
<i>Delete old record [IN]</i>	- remove only records with same pk values (must agree after pk actions execute)
<i>Setup input [IN]</i>	- set constant value (flag) on column in table from source database for each record by pk value (assumption that the pk in source table column has the same name as the in-busbar bit for the output)
<i>Update old record [IN]</i>	- if record with same pk exists in cache table, update record in target table instead insert new
<i>User SQL [PRE/IN/POST]</i>	- own user defined sql

Jeżeli potrzebujesz szczegółowej pomocy w zakresie działania poszczególnych akcji, uruchom opcję pomocy – otrzymasz okno z wyjaśnieniami

## DELETE ALL RECORDS

Przed dodaniem strumienia danych (akcja typu **PRE**), tabela docelowa zostanie wyczyszczona z wszystkich rekordów.

## DELETE ALL RECORDS FOR SOURCE

Przed dodaniem strumienia danych (akcja typu **PRE**), tabela docelowa zostanie wyczyszczona z wszystkich rekordów, jednak tylko w zakresie tych rekordów, które były standaryzowane na podstawie wskazanego źródła. Źródło jest identyfikowane na podstawie nazwy (constans) lub wskazanej wartości bitu (kolumny) magistrali wejściowej.

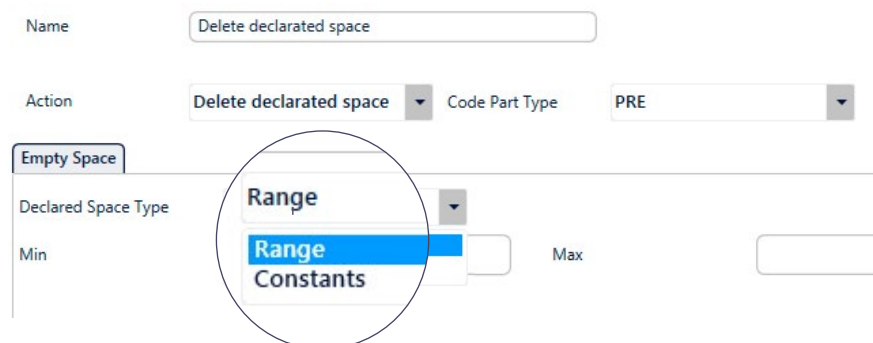
## DELETE DECLARED SPACE

Przed dodaniem strumienia danych (akcja typu **PRE**), tabela docelowa zostanie wyczyszczona z rekordów w zadeklarowanej przestrzeni.

Masz dwa sposoby, aby wskazać jaki zakres rekordów ma być skasowany:

- **RANGE**
- **CONSTANS**

Nastawa **RANGE** daje możliwość wskazania przestrzeni w zakresie wartości kolumny **PRIMARY KEY**. Rekordy w zakresie tej przestrzeni zostaną wykasowane.



Opcję zastosujesz wówczas, gdy w jednej tabeli kolekcjonujesz rekordy z różnych źródeł, a mając do dyspozycji tylko jedną kolumnę, problem unikalnego identyfikatora rozwiązałeś stwarzając przestrzenie liczbowe dla różnych źródeł (np. źródło 1: przestrzeń od 1 -> 1 000 000, źródło 2: przestrzeń -> 1 000 001 – 2 000 000 etc.).

## DELETE SPACE MIN MAX PK

Przed dodaniem strumienia danych (akcja typu **PRE**), tabela docelowa zostanie wyczyszczona z rekordów w zadeklarowanej przestrzeni. Przestrzeń do kasowania zostanie wyznaczona w obszarze pomiędzy minimalną i maksymalną wartością klucza **PRIMARY KEY** (deklarowanie klucza **PK** patrz opis **WSKAZANIE TABELI DOCELOWEJ ORAZ ZASAD DODAWANIA**).



Uwaga: jeżeli kolumny należące do PK podlegały standaryzacji, (a więc miały zmienianą wartość), GRAVITY automatycznie to wykryje i bezbłędnie zidentyfikuje właściwy rekord.

### DELETE OLD RECORD

Akcja specyficzna (bliźniacza do akcji **UPDATE OLD RECORD**), przeprowadzana dla każdego rekordu strumienia danych (w tym sensie jest to więc akcja typu **IN**). Jeżeli ustawiłeś filtr wejściowy **ACTION FOR THE SAME RECORD = PASS**, (patrz opis powyżej), to teraz możesz wskazać w jaki sposób proces GRAVITY ma się uporać z potencjalnym duplikatem. Jeżeli wybierzesz akcję **DELETE OLD RECORD**, to dotychczasowy rekord będzie kasowany, tak aby nowy rekord (potencjalny duplikat) mógł być dodany bez przeszkód.



Uwaga: jeżeli unikalny identyfikator strumienia danych podlegał standaryzacji (a więc miał zmienianą wartość), GRAVITY automatycznie to wykryje i bezbłędnie zidentyfikuje właściwy rekord.

Jeżeli ustawiłeś filtr wejściowy **ACTION FOR THE SAME RECORD = SKIP** to akcja **DELETE OLD RECORD** nie zostanie uruchomiona.

### UPDATE OLD RECORD

Akcja specyficzna (bliźniacza do akcji **UPDATE OLD RECORD**) przeprowadzana dla każdego rekordu strumienia danych (w tym sensie jest to więc akcja typu **IN**). Jeżeli ustawiłeś filtr wejściowy **ACTION FOR THE SAME RECORD = PASS**, (patrz opis powyżej), to teraz możesz wskazać w jaki sposób proces GRAVITY ma się uporać z potencjalnym duplikatem. Jeżeli wybierzesz akcję **UPDATE OLD RECORD**, to dotychczasowy rekord będzie nadpisywany przez nowy rekord (potencjalny duplikat).



Uwaga: jeżeli unikalny identyfikator strumienia danych podlegał standaryzacji (a więc miał zmienianą wartość), GRAVITY automatycznie to wykryje i bezbłędnie zidentyfikuje właściwy rekord.

Jeżeli ustawiłeś filtr wejściowy **ACTION FOR THE SAME RECORD = SKIP** to akcja **UPDATE OLD RECORD** nie zostanie uruchomiona.

### SETUP INPUT

Akcja typu **IN** (czyli przeprowadzana wielokrotnie, po każdym przetworzeniu kolejnego rekordu w operatorze **OUTPUT SQL**), daje możliwość edycji rekordu pobranego do magistrali przez któryś z operatorów typu **INPUT SQL**. Zastosujesz tego rodzaju akcję, jeżeli będziesz chciał np. ustawiać flagi pobrania rekordu ze źródła.

Zakres konfiguracji:

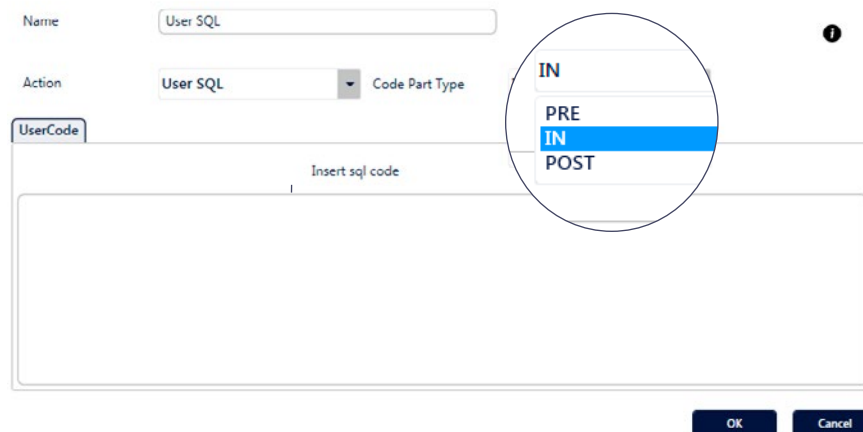
- **INPUT**.....wskaz operator typu **INPUT**, skąd pobierasz dane do magistrali
- **INPUT TABLE**.....wskaz tabelę źródłową z operatora **INPUT**, którą chcesz edytować
- **COLUMN FOR UPDATE**.....wskaz kolumnę z tabeli źródłowej, która będzie edytowana (np. miejsce ustawienia flagi)
- **VALUE FOR UPDATE**.....wskaz wartość edytowaną (np. flagi)
- **ID SOURCE RECORD**.....wskaz sposób identyfikacji rekordu (np. do ustawienia flagi); wskazujesz kolumnę unikalną oraz kolumnę z magistrali wejściowej do operatora **OUTPUT SQL**;

Operator **SETUP INPUT** daje szerokie możliwości interakcji z danymi źródłowymi. Po pobraniu rekordów, możesz zaznaczać, które rekordy zostały pobrane. Dzięki temu możesz prowadzić pobrania danych w sposób jednoznacznie inkrementacyjny (ustawiając warunek pobrania do wybranej wartości flagi w operatorze **INPUT**).

## USER SQL

Akcja, która daje Tobie wolną rękę. Możesz stworzyć dowolny kod **SQL** oraz wskazać moment jego zastosowania (patrz ilustracja poniżej):

- **PRE** (przed akcją dodania strumienia danych)
- **POST** (po akcji dodania strumienia danych)
- **IN** (w trakcie dodawania strumienia danych-uwaga: kod wówczas jest wywoływany wielokrotnie, po dodaniu każdego rekordu strumienia danych)



Podczas pisania własnego kodu **SQL**, możesz korzystać z wartości sygnału na magistrali wejściowej do operatora **OUTPUT SQL**. Do wartości kolumny strumienia danych odwołujesz się poprzez znak prowadzący '@' oraz nazwę bitu (kolumny) w strumieniu danych wejściowych (np.@id).

## DEKLARACJA KODU DODAWANIA STRUMIENIA DANYCH: WSKAZANIE TABELI DOCELOWEJ ORAZ ZASAD WPROWADZANIA STRUMIENIA DANYCH

### KROK 1

Po prawej stronie okna, widzisz wszystkie tabele umiejscowione w bazie docelowej, wskazanej w **PHISICAL LINK**.

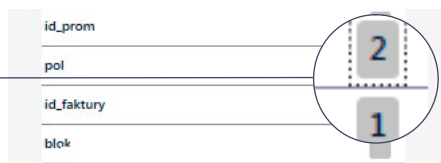


Wybór tabeli docelowej odbywa się poprzez uchwycenie właściwej tabeli i upuszczenie na obszarze roboczym.

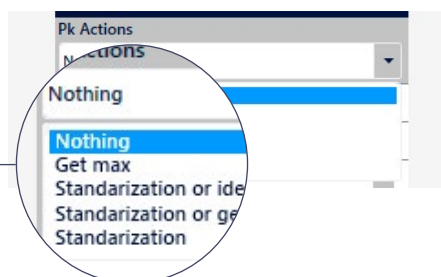


Na obszarze roboczym możesz położyć kilka tabel, jako tabele docelowe, jednak musisz je powiązać, wskazując relację rodzic – dziecko. W przypadku, gdy położysz kilka tabel niezależnie (nie wskazując relacji), proces w trakcie przetwarzania zostanie zatrzymany.

Po wskazaniu tabeli 'celu', możesz przystąpić do deklarowania zasad umieszczenia strumienia danych w tabeli.

**KROK 2**

Wskaż teraz pola **PRIMARY KEY** (klucz **PRIMARY KEY** to szereg kolumn jednoznacznie identyfikujących rekord), poprzez zaznaczenie wybranych kolumn.

**KROK 3**

Powinieneś teraz określić, jak proces ma się zachować względem pól **PRIMARY KEY** podczas dodawania rekordu (opcja dostępna w belce nagłówka tabeli).

Masz do wyboru następujące akcje dla pól **PRIMARY KEY**:

- **NOTHING**
- **GET MAX**
- **STANDARIZATION OR IDENTITY**
- **STANDARIZATION OR GET MAX**
- **STANDARIZATION**

Zanim dokładnie opiszemy każdą z akcji z osobna, najpierw sformułujemy definicję standaryzacji, w kontekście operatora **OUTPUT SQL**.

**CO TO JEST STANDARYZACJA?**

Standaryzacja to proces zamiany wartości pola (kolumny) na wartość standardową, na podstawie słownika kojarzącego wartość dotychczasową z wartością standardową.

W kontekście operatora **OUTPUT SQL** słownik standaryzujący powstaje automatycznie -ad hoc- podczas procesu przebiegającego przez operator **OUTPUT SQL** w przypadku, gdy napotkamy sytuację zamiany wartości pól (kolumn) **PRIMARY KEY**.

Słownik standaryzujący jest tworzony przez **GRAVITY** automatycznie i zapamiętany w bazie docelowej.

Słownik standaryzujący jest tworzony dla tabeli 'celu'.

**DLACZEGO STANDARYZUJEMY?**

Strumień danych składowany w tabeli 'celu', może mieć swoje źródło w wielu niezależnych bazach zasilających. W kontekście tego faktu, aby zeszkładować wszystkie dane w jednej bazie, będziesz musiał podjąć decyzję o zmianie wartości pól unikalnych (jednoznacznie identyfikujących), ponieważ w różnych źródłach te wartości mogą być takie same.

**JAK PRZEBIEGA STANDARYZACJA?**

Technicznie, jest to automatyczna akcja tworzenia słownika przekodowań. Słownik przekodowań (słownik standaryzujący) jest związany z tabelą 'celu'. Przekodowanie łączy z jednej strony identyfikator źródła (bit źródła) oraz wartości pól starego PK (identyfikatory unikalne ze źródła) z identyfikatorami unikalnymi nowoutworzonymi w tabeli 'celu'

Jeżeli chcesz zobaczyć jak wyglądają rekordy w słownikach standaryzujących, wybierz opcję **SHOW CACHE TABLES**.



Jeżeli już wiesz czym jest standaryzacja, przeczytaj koniecznie poniższy opis zastosowania każdej z akcji względem pól **PRIMARY KEY**.



#### NOTHING

Pola PRIMARY KEY (jednoznacznie identyfikujące) pozostają bez zmian.

#### GET MAX

Proces oblicza nową wartość dla ostatniego pola PRIMARY KEY poprzez inkrementację ostatniego zapisu w tabeli docelowej. Słownik standaryzujący nie jest budowany.

#### STANDARIZATION OR IDENTITY

Proces w pierwszym kroku sprawdza, czy pola nie podlegały wcześniejszej standaryzacji. Sprawdzenie polega na badaniu występowania pierwotnej (źródłowej) wartości pól klucza Primary Key w słowniku standaryzującym. Jeżeli wartość występuje, następuje zamiana wartości źródłowej na wartość standardową (czyli wartość ze słownika standaryzującego). Jeżeli wartość źródłowa nie występuje w słowniku standaryzującym następuje akcja pobrania ID nadanego przez bazę celu oraz umieszczenie nowej wartości w słowniku standaryzującym.

Jeżeli wybierzesz tę akcję będziesz musiał wskazać identyfikator źródła, wskazując odpowiedni bit magistrali wejściowej (jeżeli masz taki bit) lub wpisując nazwę własną, stosowaną zawsze dla tego źródła danych.

#### STANDARIZATION OR GET MAX

Proces w pierwszym kroku sprawdza, czy pola nie podlegały wcześniejszej standaryzacji. Sprawdzenie polega na badaniu występowania pierwotnej (źródłowej) wartości pól klucza Primary Key w słowniku standaryzującym. Jeżeli wartość występuje, następuje zamiana wartości źródłowej na wartość standardową (czyli wartość ze słownika standaryzującego). Jeżeli wartość źródłowa nie występuje w słowniku standaryzującym następuje akcja GET MAX (patrz opis akcji GET MAX) oraz umieszczenie nowej wartości w słowniku standaryzującym.

Jeżeli wybierzesz tę akcję będziesz musiał wskazać identyfikator źródła wskazując odpowiedni bit magistrali wejściowej (jeżeli masz taki bit) lub wpisując nazwę własną stosowaną zawsze dla tego źródła danych.

#### STANDARIZATION

Akcja specyficzna. Podczas procedowania system sprawdza, czy w słowniku standaryzującym jest już wartość PRIMARY KEY. Jeżeli nie ma, to system uzupełnia słownik standaryzujący oraz dodaje rekord. W przeciwnym wypadku (wartość pola Primary Key jest już w słowniku standaryzującym) system nie dodaje nowego rekordu, tylko stosuje polecenie UPDATE (zmiana) względem wcześniej dodanego identycznego rekordu.

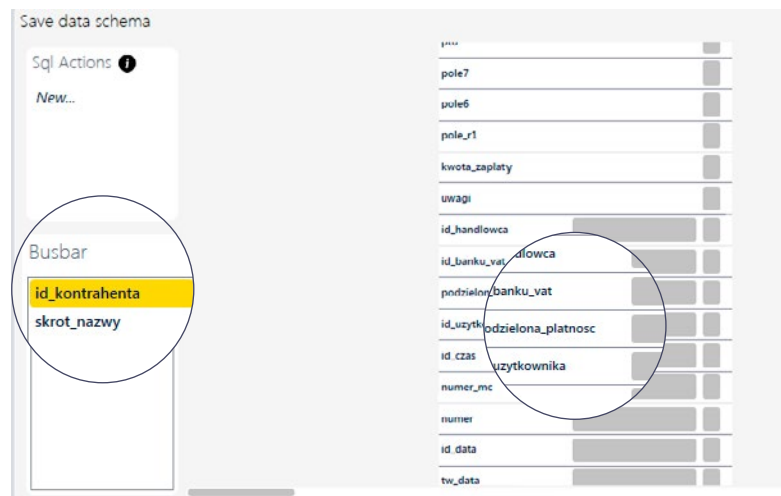
Jeżeli wybierzesz tę akcję będziesz musiał wskazać identyfikator źródła, wskazując odpowiedni bit magistrali wejściowej (jeżeli masz taki bit) lub wpisując nazwę własną, stosowaną zawsze dla tego źródła danych.

## KROK 4

Teraz powinieneś wskazać, jak sygnał wejściowy na magistrali wejściowej przekłada się na poszczególne pola (kolumny) tabeli celu. Możesz to uczynić w następujący sposób:

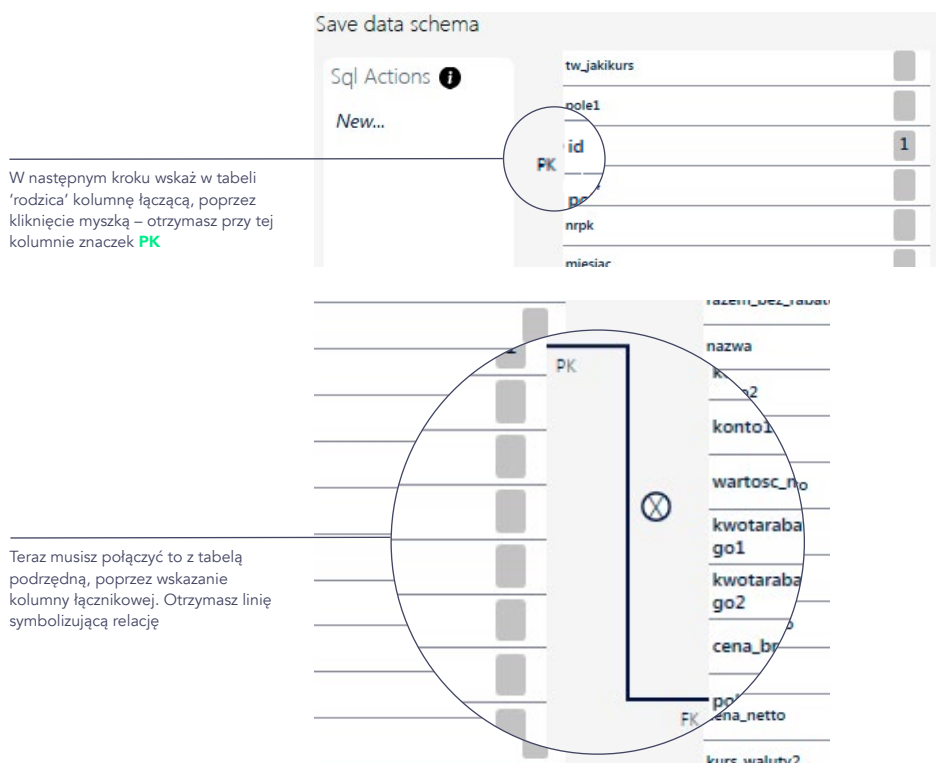
- Wybierz kolumnę magistrali wejściowej (patrz na ilustracji poniżej – zaznaczony obszar po Twojej lewej stronie)
- System automatycznie wskaże jakie pola są możliwe do przyrównania poprzez wyszarzenie (patrz ilustracja poniżej – obszar zaznaczony po Twojej prawej stronie); pola te są wybierane na podstawie zgodności typów danych;
- Wybierz pole tabeli, poprzez klik we właściwą kolumnę tabeli celu
- Wybierz następną kolumnę magistrali wejściowej i powtórz czynności





### KROK 5

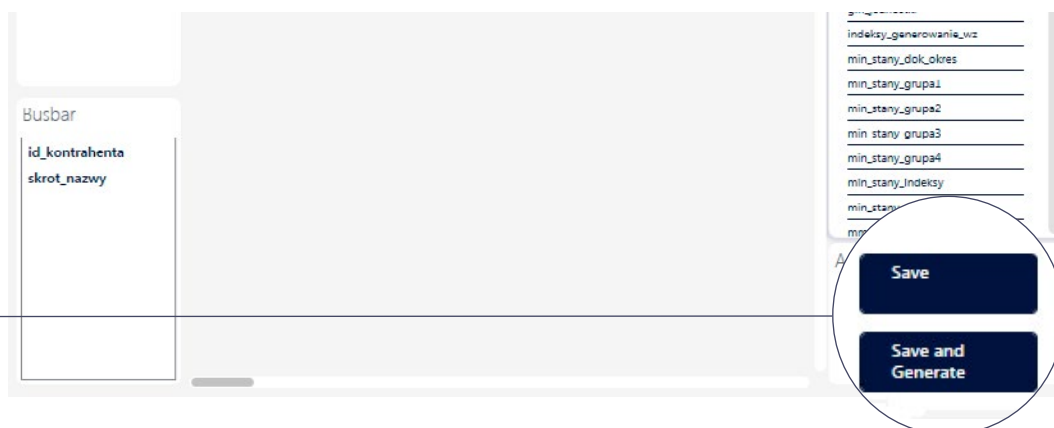
Jeżeli chcesz, aby strumień danych magistrali wejściowej obsługiwał również tabelę podrzędną względem tabeli głównej, możesz to uczynić w następnym kroku. Wybierz tabelę relacyjną i umieść ją na obszarze roboczym – akcja identyczna jak w przypadku tabeli głównej: należy tabelę relacyjną uchwycić na liście dostępnych tabel (lista w obszarze po lewej stronie) oraz upuścić na obszarze roboczym.



W przypadku standaryzacji pól łącznikowych, system automatycznie zadba o standaryzację pól, również w tabelach podrzędnych.

Możesz powtarzać ten krok dowolną liczbę razy, a liczba zagłębień relacji podrzędnych może być dowolna.

## DEKLARACJA KODU DODAWANIA STRUMIENIA DANYCH: ZAPAMIĘTANIE KONFIGURACJI



Jeżeli ustawiłeś wszystkie akcje oraz zadeklarowałeś tabele wejścia strumienia danych, musisz swój diagram zapamiętać.

Jeżeli posiadasz doświadczenie – możesz go edytować, jednak nie jest to niezbędne: GRAVITY generuje go samodzielnie, na podstawie zadeklarowanych akcji.

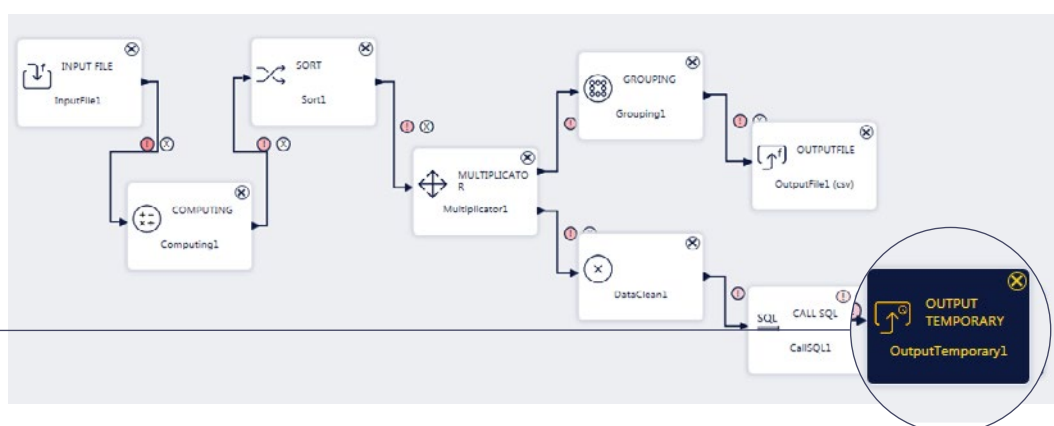


Podczas zapamiętywania GRAVITY automatycznie generuje kod w formacie XML. Możesz go zobaczyć w zakładce **INSERT CODE**.

Powyżej opisaliśmy, w jaki sposób korzystać z operatora **OUTPUT SQL**. Zdajemy sobie sprawę, że to jeden z najbardziej złożonych komponentów GRAVITY, dlatego – dla lepszego zobrazowania działania operatora **OUTPUT SQL** – zachęcamy do zapoznania się z przykładami (patrz rozdział **PRZYKŁADY ZASTOSOWANIA**). Operatorowi **OUTPUT SQL** oprócz zwykłych przykładów kontekstowych poświęciliśmy osobny akapit **ZASTOSOWANIE OPERATORA OUTPUT SQL**.

### 8.2.7 OUTPUT TEMPORARY

- i** Operator **OUTPUT TEMPORARY** ma dwa zastosowania:
- zaślepienie przetwarzania danych w aplikacji GRAVITY
  - gromadzenie danych strumienia wejściowego do wykorzystania dla operatora **STANDARIZATION**



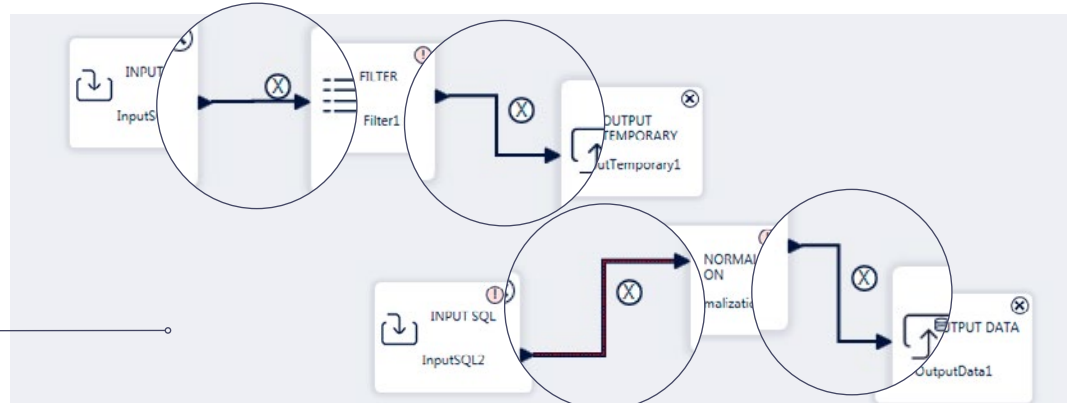
Przykład projektu GRAVITY z użyciem operatora **OUTPUT TEMPORARY**.

## 8.3 OPERATORY PRZETWARZANIA

### 8.3.1 BUSBAR

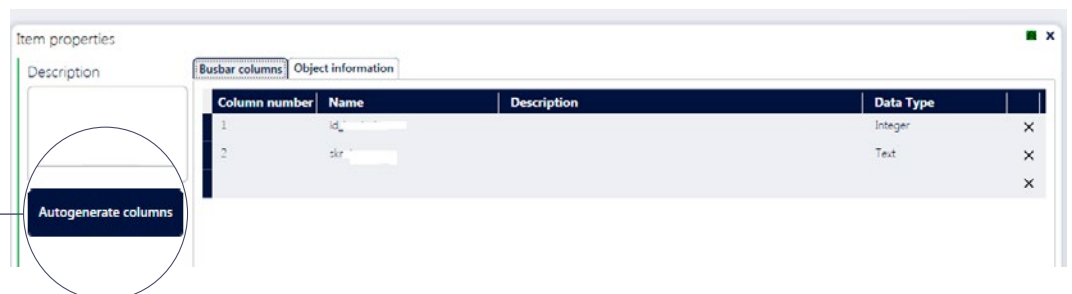


BUSBAR to magistrala łącząca dwa operatory. Po magistralach płyną strumienie przetwarzanych danych.



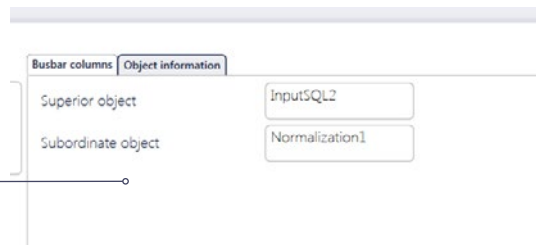
Przykład magistral.

### KONFIGURACJA MAGISTRALI



Możesz wprowadzić samodzielnie opisy do kolumn, lub raz jeszcze wygenerować kolumny na podstawie operatora źródłowego (patrz opcja **AUTOGENERATE**).

Zasadniczą cechą magistrali jest jej szerokość, czyli kolumny transportujące dane w strumieniu danych. Po połączeniu dwóch operatorów, magistrala jest automatycznie konfigurowana na podstawie sygnału operatora źródłowego.



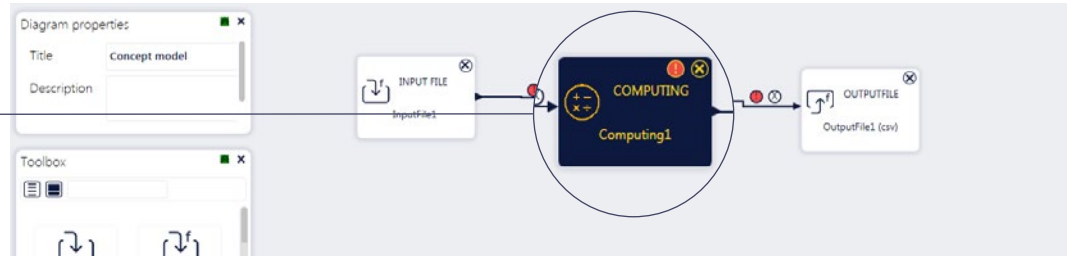
W zakładce **OBJECT INFORMATION** możesz zobaczyć nazwy operatora źródłowego oraz operatora docelowego.

### 8.3.2 COMPUTING

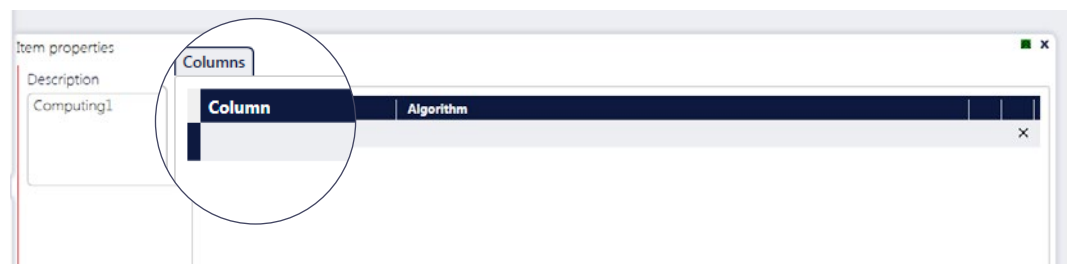


Operator COMPUTING przetwarza strumień danych znajdujących się na wejściu operatora. Operacja przetwarzania odbywa się dla każdego rekordu strumienia danych. Wynik przetwarzania, to strumień danych na wyjściu operatora.

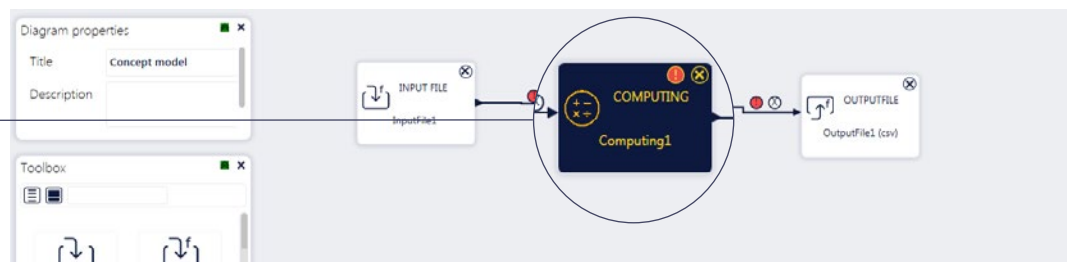
Przykład projektu GRAVITY z użyciem operatora **COMPUTING**.



### KONFIGURACJA ALGORYTMÓW PRZETWARZANIA STRUMIENIA DANYCH



Możesz przetworzyć dowolną kolumnę strumienia danych, znajdujących się na wejściu operatora. Wystarczy, że wskażesz kolumnę oraz zredagujesz algorytm przetwarzania.



Edytor pomaga Tobie: każdą z funkcji możesz wybrać ze słownika.



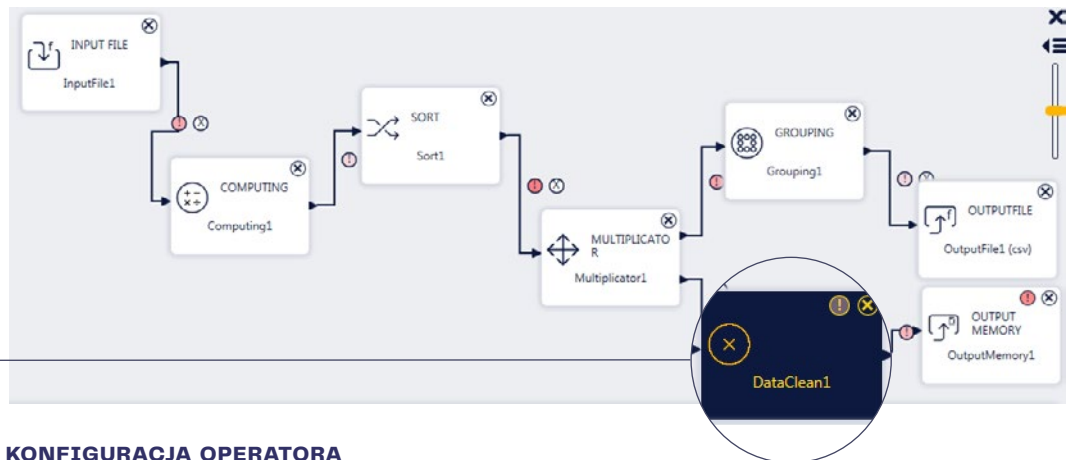
W algorytmie możesz wykorzystać funkcje:

- **GET\_VALUE** .... wartość wskazanej kolumny magistrali
- **GET\_AGGR** ....suma wartości wskazanej kolumny magistrali, dla wszystkich danych na wejściu
- **GET\_OUTTEMP\_AGGR** ....suma wartości kolumny danych zgromadzonych w zbiorze **OUTPUT\_TEMPORARY**

### 8.3.3 DATA CLEAN

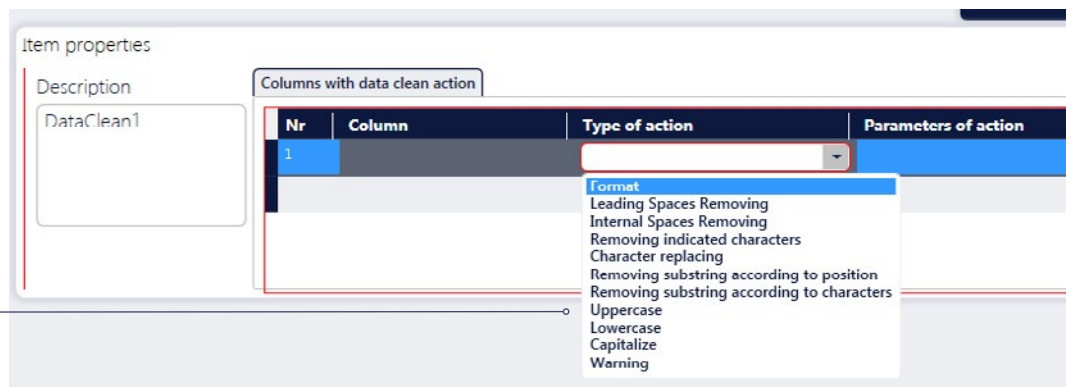


Operator DATA CLEAN umożliwia przetwarzanie każdego rekordu strumienia znajdującego się na magistrali wejściowej, dla wskazanych kolumn wg zasad wybranych przez projektanta.



Przykład projektu GRAVITY z użyciem operatora **DATA CLEAN**.

### KONFIGURACJA OPERATORA



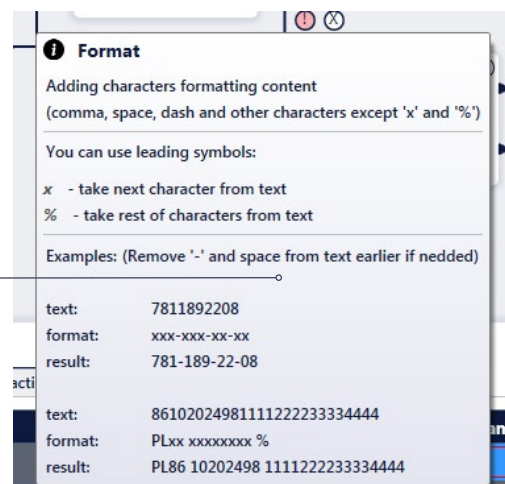
Nr	Column	Type of action	Parameters of action
1		Format	

Format

- Leading Spaces Removing
- Internal Spaces Removing
- Removing indicated characters
- Character replacing
- Removing substring according to position
- Removing substring according to characters
- Uppercase
- Lowercase
- Capitalize
- Warning

Każda kolumna magistrali może podlegać akcjom opisanym poniżej.

### FORMAT



**Format**

Adding characters formatting content (comma, space, dash and other characters except 'x' and '%')

You can use leading symbols:

x - take next character from text  
% - take rest of characters from text

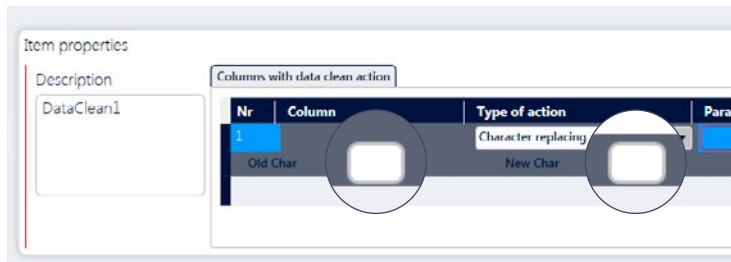
Examples: (Remove '-' and space from text earlier if needed)

text:	7811892208
format:	xxx-xxx-xx-xx
result:	781-189-22-08
text:	86102024981111222233334444
format:	PLxx xxxxxxxx %
result:	PL86 10202498 1111222233334444

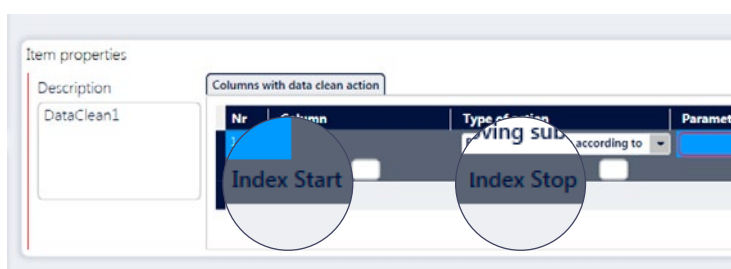
Formatowanie danych dla wskazanej kolumny zgodnie z zasadami.

- **LEADING SPACE REMOVING** .... Usuwanie spacji początkowych.
- **INTERNAL SPACE REMOVING** .... Usuwanie spacji występujących pomiędzy innymi znakami.
- **REMOVING INDICATED CHARACTERS** .... Usuwanie z kolumny wskazanego znaku.

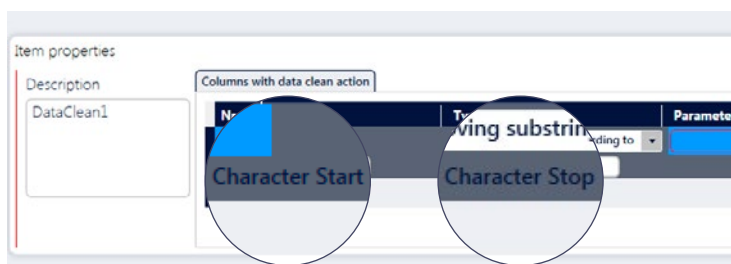
- **CHARACTER REPLACING** .... Zmiana znaku na inny.



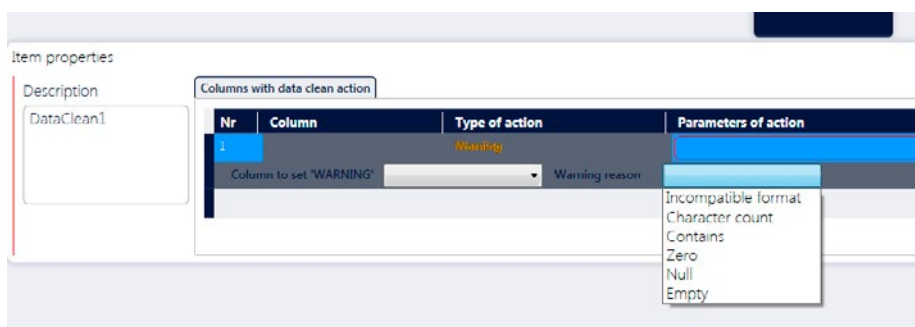
- **REMOVING SUBSTRING ACORDING TO POSTITION** .... Usuwanie ciągu znaków ze wskazanej pozycji. Konfigurujesz kolejny numer początku oraz końca usuwanego ciągu..



- **REMOVING SUBSTRING ACORDING TO CHARACTERS** .... Usuwanie ciągu znaków. Konfigurujesz znak początkowy oraz znak końcowy ciągu do usunięcia.



- **UPPERCASE** .... Po akcji, wszystkie znaki wskazanej kolumny są duże
- **LOWERCASE** .... Po akcji, wszystkie znaki wskazanej kolumny mają małe znaki.
- **CAPITALIZE** .... Po akcji, i pierwszy znak jest z wielkiej litery.
- **WARNING** .... Akcja generowania ostrzeżenia. Podczas konfiguracji wskazujemy kolumnę badaną, przyczynę wywołającą ostrzeżenie (patrz słownik w ilustracji poniżej) oraz kolumnę strumienia danych, w której należy uplasować potencjalne ostrzeżenie.

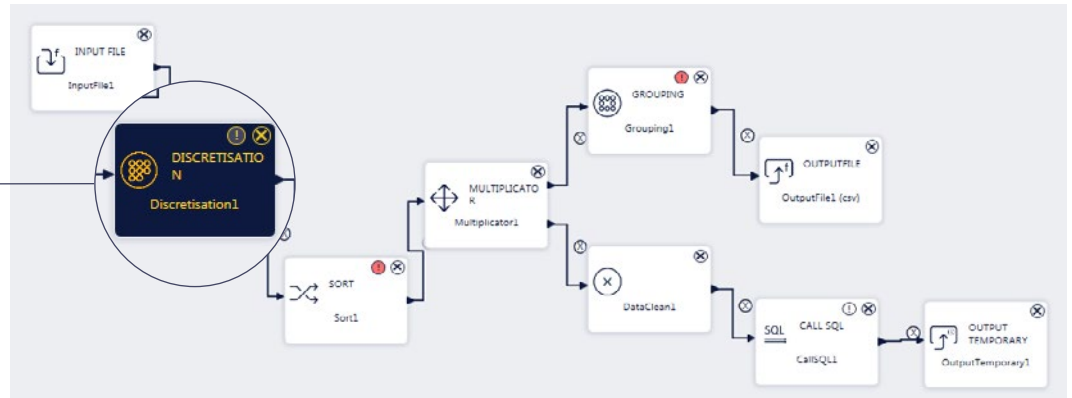


### 8.3.4 DISCRETISATION

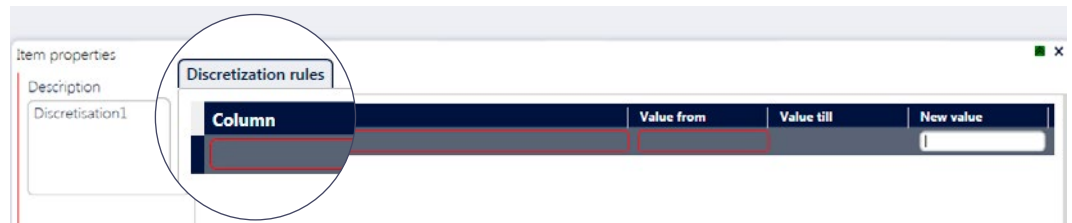


Operator DISCRETISATION umożliwia przetworzenie wartości danych w przetwarzanym strumieniu z wartości ciągłych na wartości dyskretne. Oznacza to, że operator zamienia wartości należące do przedziału danych na jedną wartość (wartość dyskretną) charakterystyczną dla danego przedziału.

Przykład projektu GRAVITY z użyciem operatora **DISCRETISATION**.



#### KONFIGURACJA OPERATORA



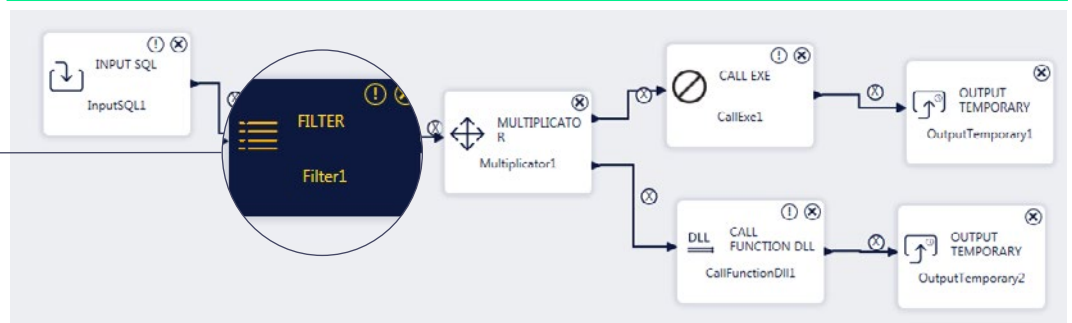
Użytkownik może podać procesowi dyskretyzacji wybraną kolumnę, znajdującą się na magistrali wejściowej. Wynikiem działania operatora (w magistrali wyjściowej) jest strumień danych, w którym dokonano procesów dyskretyzacji wybrane kolumny.

### 8.3.5 FILTER



Operator FILTER umożliwia przefiltrowanie strumienia danych z magistrali wejściowej. Magistrala na wyjściu operatora zawiera strumień danych przefiltrowanych.

Przykład projektu GRAVITY z użyciem operatora **FILTER**.



#### USTAWIENIE WARUNKÓW FILTROWANIA

##### Wariant prosty

W tym wariantcie możesz skorzystać z parametrów zebranych w słowniku: każdej kolumny strumienia danych magistrali na wejściu. Wystarczy, że wybierzesz kolumnę i zadeklarujesz warunek. Jeżeli wskażesz więcej warunków: pomiędzy warunkami będzie zachodziła interakcja **AND**.

Na podstawie wariantu prostego, system sam stworzy bardziej wyrafinowany kod, dostępny w następnym zakładce.

### Wariant złożony

Jeżeli chcesz stworzyć bardziej złożony warunek filtrowania, możesz posłużyć się kompilowanym kodem w zakładce następniej.



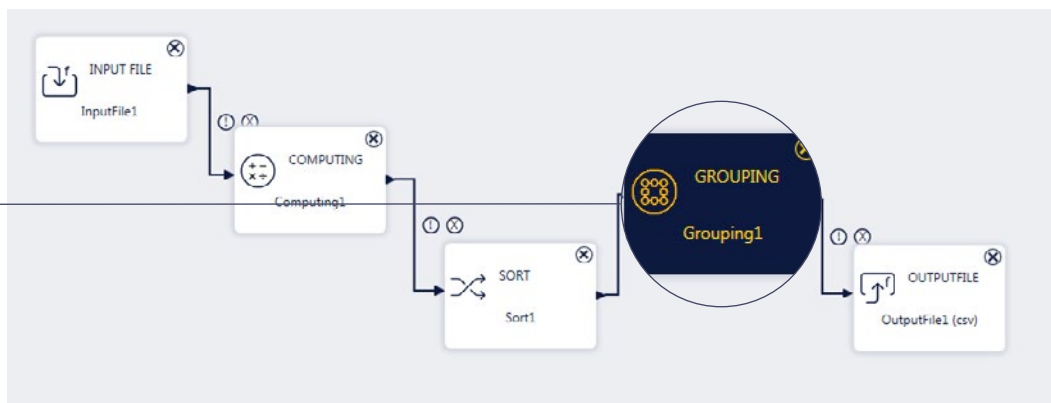
W opcji możesz skorzystać z parametrów zebranych w słowniku: każdej kolumny strumienia danych magistrali na wejściu oraz z funkcji agregujących:

- **GET\_AGGR** ....suma wartości wskazanej kolumny magistrali, dla wszystkich danych na wejściu
- **GET\_OUTTEMP\_AGGR** ....suma wartości kolumny danych zgromadzonych w zbiorze **OUTPUT\_TEMPORARY**

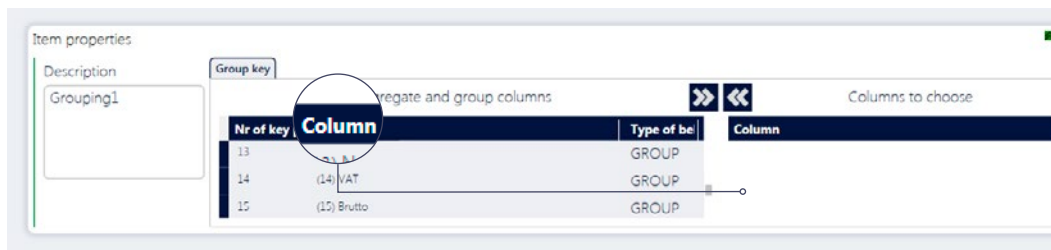
### 8.3.6 GROUPING

**i** Operator **GROUPING** umożliwia przetworzenie strumienia danych na wejściu, w ten sposób, że dla wskazanego klucza grupującego (sekwencja dowolnej liczby wybranych kolumn) zaczyna obowiązywać zasada unikalności wartości (braku powtórzeń). Oznacza to, że sekwencja wartości kolumn grupujących może wystąpić tylko raz. Dla kolumn nie znajdujących się w kluczu grupującym, należy wskazać funkcję agregacyjną (czyli jak ma zachować się akcja przetwarzania dla kolumny w przypadku wystąpienia powtórzeń).

Przykład projektu GRAVITY z użyciem operatora **GROUPING**.



### KONFIGURACJA ZASAD GRUPOWANIA



Możesz ustawić klucz sortujący, składający się z dowolnej liczby kolumn magistrali poprzez akcję drag and drop (chwytną kolumnę magistrali wchodzącej i upuszczasz na listę klucza grupującego). Ponadto, dla pozostałych kolumn magistrali wchodzącej określasz funkcję agregacyjną (**TYPE OF BEHAVIOUR**).

Na wyjściu operatora będziesz miał strumień danych przetworzonych o tej samej szerokości (czyli tej samej liczbie kolumn).



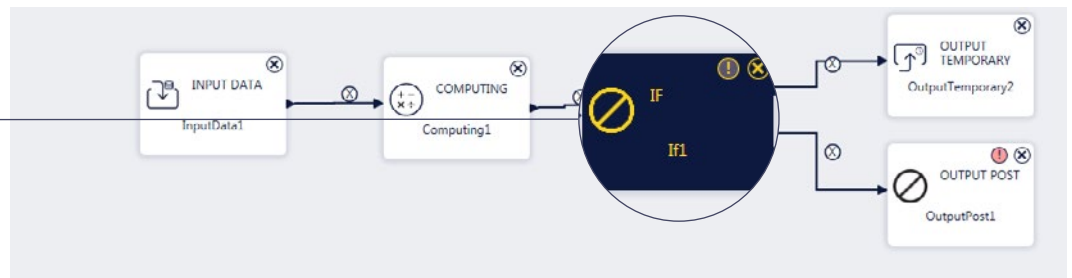
## 8.3.7 IF



Operator IF daje możliwość skierowania całego strumienia danych w wybranym kierunku (masz do wyboru dwa kierunki), w zależności od zadeklarowanego przez Ciebie warunku. Jeżeli chcesz przeprowadzić wybór wielowariantowy, możesz umieścić szeregowo tyle operatorów IF, ile potrzebujesz do realizacji swojego zadania.

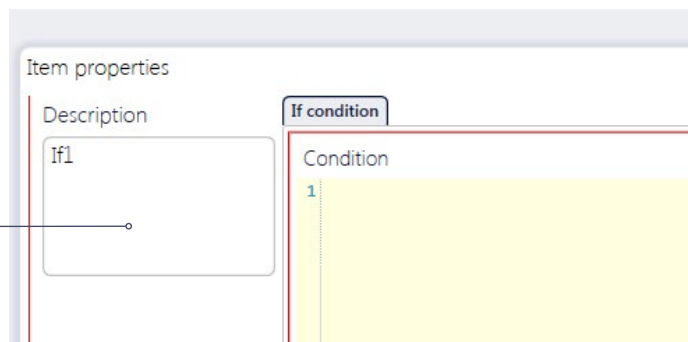
## KONFIGURACJA OPERATORA

Przykład projektu GRAVITY z użyciem operatora IF.



Podczas przetwarzania, jeżeli warunek będzie spełniony, strumień danych znajdzie się na magistrali wyjściowej górnej, w przypadku zaś niespełnienia warunku, strumień danych popłynie magistralą dolną.

W operatorze masz możliwość zadeklarowania warunku, w oparciu o strumień danych wejściowych



## Funkcje dostępne

- **GET\_AGGR** ....suma wartości wskazanej kolumny magistrali, dla wszystkich danych na wejściu

Podczas tworzenia warunku, masz możliwość odwoływania się do funkcji agregujących – wszystkie są dostępne w słowniku.

DisplayName	Category	Description
GET_AGGR	methods	Aggregation values of incoming column
GET_OUTTEMP_AGGR	methods	Aggregation values of output temporary col
function	columns	Value of column 'function'
id	columns	Value of column 'id'
param1	columns	Value of column 'param1'
param2	columns	Value of column 'param2'

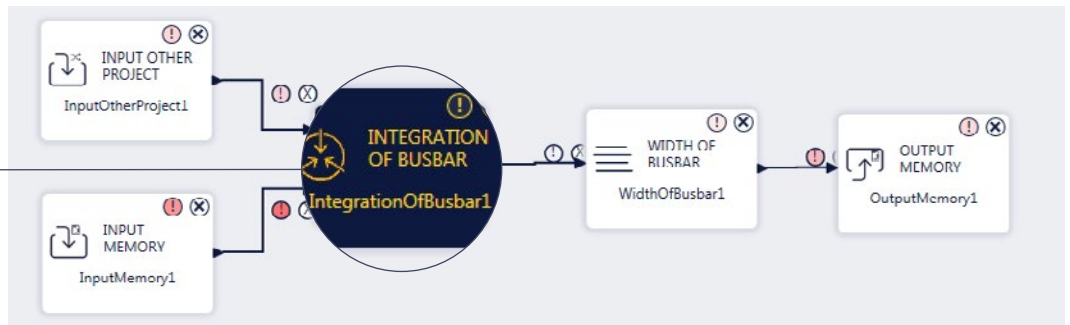
- **GET\_OUTTEMP\_AGGR** ....suma wartości kolumny danych, zgromadzonych w zbiorze OUTPUT\_TEMPORARY

### 8.3.8 INTEGRATION OF BUSBAR



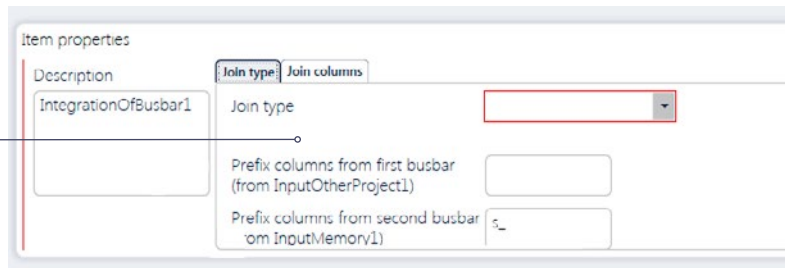
Operator INTEGRATION OF BUSBAR umożliwia łączenie dwóch strumieni danych, zgodnie z regułami wskazanymi przez projektanta aplikacji GRAVITY. Magistrala na wyjściu operatora ma szerokość (liczbę kolumn) równą sumie szerokości magistral wejściowych.

Przykład projektu GRAVITY z użyciem operatora **INTEGRATION OF BUSBAR**.



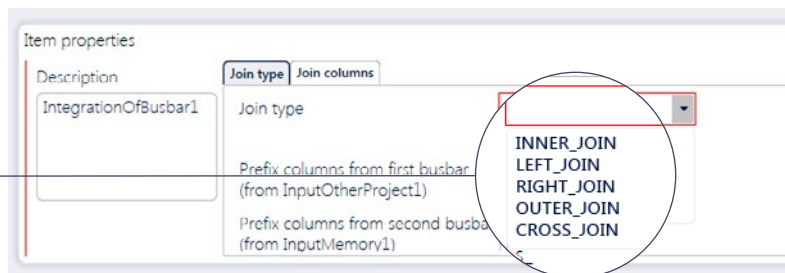
#### KONFIGURACJA OPERATORA

Możesz wskazać prefiksy charakterystyczne dla źródła wejściowego, którymi będą oznaczone kolumny wyjściowego strumienia danych.



Na magistrali wyjściowej otrzymamy strumień danych, którego szerokość (liczba kolumn) jest zależna od akcji jaką zlecimy operatorowi. Z całą pewnością, będziesz chciał mieć precyzyjny opis każdego bitu (kolumny) strumienia danych.

Masz do wyboru akcje charakterystyczne dla języka **SQL** (patrz słownik):



Konfigurując integrację dwóch magistral, będziesz musiał podjąć decyzję, na jakich zasadach chcesz połączyć rekordy z dwóch strumieni danych.

- **INNER JOIN** .... Za pomocą tej akcji połączysz ze sobą te rekordy z dwóch magistral, które mają równe wartości dla kolumn łączących (definicja kolumn łączących patrz ilustracja poniżej). Liczba rekordów na magistrali wychodzącej jest iloczynem wszystkich dopasowań kolumn łączących (część wspólna). Rekordy z obydwóch strumieni, których nie można połączyć są odfiltrowane.
- **LEFT JOIN** .... Dla tej akcji przyjmujemy jako nadrzędną magistralę górną (wchodzi do operatora na górze).  
Na magistralę wyjściową trafiają:
  - » wszystkie rekordy dopasowane (patrz akcja **INNER JOIN**). Łączenie odbywa się wg kolumn łączących (patrz ilustracja poniżej).
  - » wszystkie rekordy magistrali nadrzędnej niedopasowane poszerzone o kolumny magistrali podrzędnej z wartościami **NULL**.

- **RIGHT JOIN** .... Dla tej akcji, przyjmujemy jako nadrzędną magistralę dolną (wchodzi do operatora na dole). Na magistralę wyjściową trafiają:
  - » wszystkie rekordy dopasowane (patrz akcja **INNER JOIN**). Łączenie odbywa się wg kolumn łączących (patrz ilustracja poniżej).
  - » wszystkie rekordy magistrali nadrzędnej niedopasowane, poszerzone o kolumny magistrali podrzędnej z wartościami **NULL**.
- **OUTER JOIN** .... Podczas tej akcji operator przepuści wszystkie rekordy, które podczas akcji **LEFT JOIN** nie byłyby łączone (mają **NULL** na magistrali poszerzonej) oraz przepuści wszystkie rekordy, które podczas akcji **RIGHT JOIN** nie byłyby łączone (mają **NULL** na magistrali poszerzonej) oraz wszystkie rekordy, które podczas akcji **LEFT JOIN** (lub **RIGHT JOIN**) byłyby połączone z rekordami magistrali podrzędnej (część wspólna).
- **CROSS JOIN** .... Na magistrali wyjściowej popłynie strumień danych, będący iloczynem kartezyjskim jednego i drugiego strumienia (macierz).

W zakładce **JOIN COLUMNS** wskażesz kolumny łączące

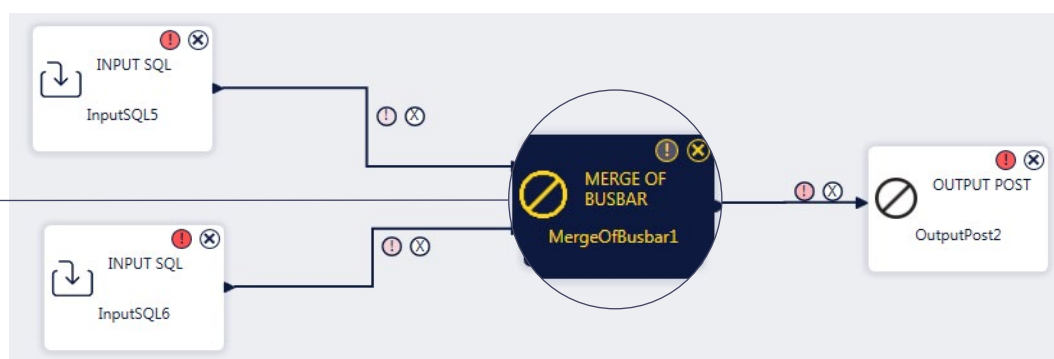


### 8.3.9 MERGE OF BUSBAR

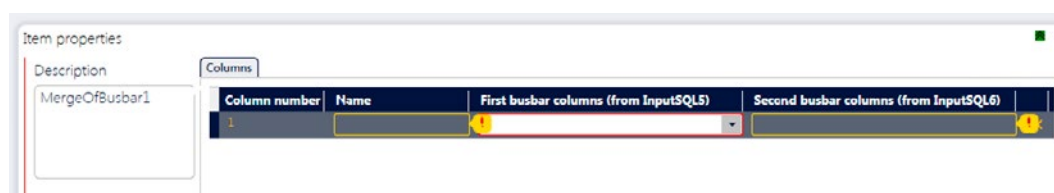


Operator **MERGE OF BUSBAR** umożliwia łączenie dwóch strumieni danych o takiej samej naturze. Strumień danych na wyjściu operatora jest równy sumie długości dwóch strumieni danych wejściowych, zaś jego szerokość jest równa szerokości strumienia szerszego z dwóch strumieni łączonych.

Przykład projektu GRAVITY z użyciem operatora **MERGE OF BUSBAR**.



#### KONFIGURACJA OPERATORA



Każdy bit magistrali wyjściowej powinien być zadeklarowany, poprzez wskazanie bitów z magistrali wchodzących (magistrali strumieni łączonych).



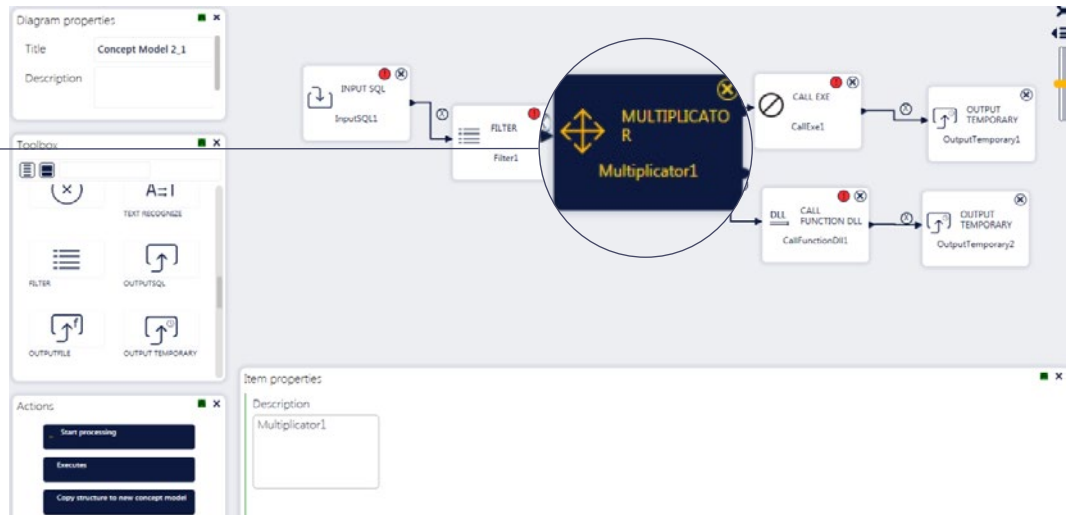
Długość strumienia (nie mylić z szerokością) wyjściowego będzie równy sumie długości strumieni wchodzących do operatora.

### 8.3.10 MULTIPLICATOR



Operator MULTIPLICATOR umożliwia przetworzenie magistrali na wejściu, na dwie niezależne magistrale zawierające identyczny strumień danych.

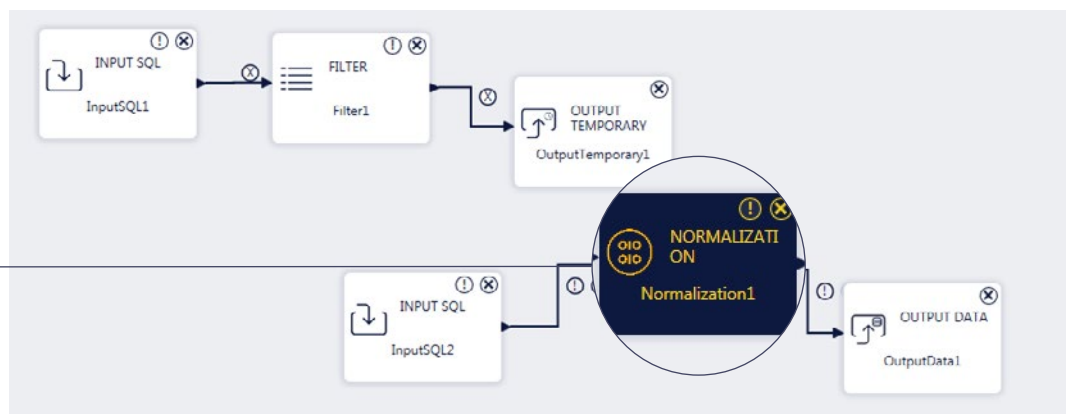
Przykład projektu GRAVITY z użyciem operatora MULTIPLICATOR.



### 8.3.11 NORMALIZATION

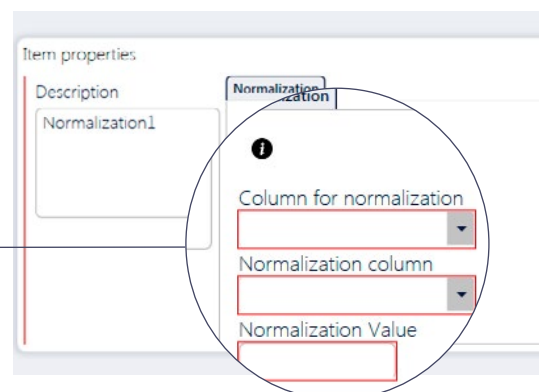
Operator NORMALIZATION umożliwia sprowadzenie danych do wspólnej jednostki. W trakcie akcji sprowadzania do wspólnej jednostki, dane są przeliczane według przelicznika zawartego w zbiorze zewnętrznym, zawierającym przeliczniki.

Przykład projektu GRAVITY z użyciem operatora NORMALIZATION.



### KONFIGURACJA OPERATORA

Konfigurując operator musisz wskazać



- Kolumnę z magistrali wejściowej, podlegającą normalizacji (**COLUMN FOR NORMALIZATION**) (ta kolumna będzie przekształcana)
- Kolumnę normalizującą (**NORMALIZATION COLUMN**) (kolumna zawierająca miarę)
- Nazwę jednostki normalizującej (**NORMALIZATION VALUE**) (opis uniwersalnej jednostki)

Item properties

Description: Normalization1

Normalization

Column for normalization: [dropdown]

Normalization column: [dropdown]

Normalization Value: [input]

Output temporary object: OutputTemporary1

Output temporary converter column: id\_kontrahenta

Output temporary normalization column: [dropdown]

Konfigurując operator musisz wskazać

Konfigurując operator, musisz wskazać sposób konwersji (patrz ilustracja powyżej):

- Dane zewnętrzne, na podstawie których dokonamy konwersji (dane są zawarte w operatorze **OUTPUT TEMPORARY** jako wyniku innego strumienia przetwarzania)
- Kolumnę z **OUTPUT TEMPORARY** (**OUTPUT TEMPORARY NORMALIZATION COLUMN**) (kolumna zawierająca miarę; łączenie z danymi strumienia wejściowego magistrali operatora)
- Kolumnę z **OUTPUT TEMPORARY** (**OUTPUT TEMPORARY CONVERTER COLUMN**) (kolumna mnożnik konwersji; dana wyjściowa jest iloczynem danej wejściowej i mnożnika)

**Normalization**

An example of normalization is the reconciliation of units:

**Column For Normalization** = column with value to change [quantity].

**Normalization Column** = column with units [unit of measure].

**Normalization Value** = new unified value normalization column - 'kg'.

**Output Temporary Normalization Column** = column in OT with unit of measure [unit].

**Output Temporary Column Result** = column in OT with conversion factor to wanted unit of measure (kg) [converter].

Przykład akcji normalizacyjnej

In Data			Output temporary (dictionary with conversion to kg)		Out Data		
Index	Quantity	Unit of measure	Unit	Converter	Index	Quantity	Unit of measure
1	3	kg	kg	1	1	3	kg
2	1600	g	g	0.001	2	1.6	kg
3	3	pack	pack	0.25	3	0.75	kg

### 8.3.12 OCR

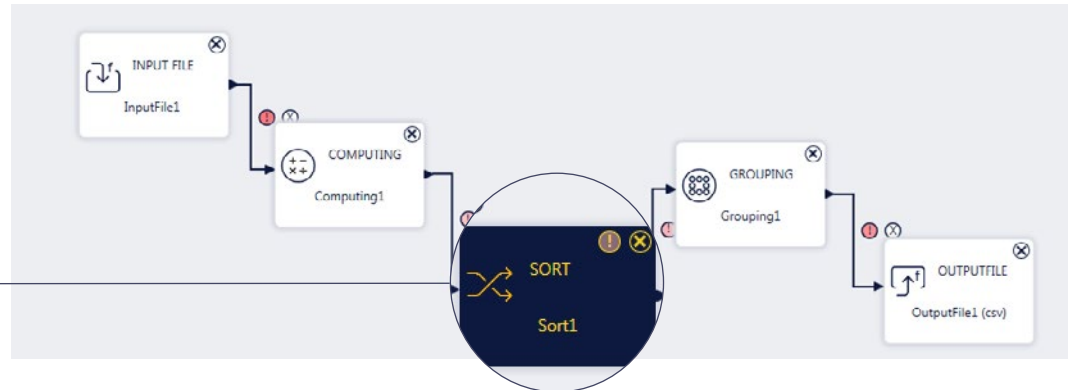


Operator OCR zamienia obiekt graficzny na strumień tekstu, poprzez rozpoznanie tekstu w obiekcie graficznym.

### 8.3.13 SORT



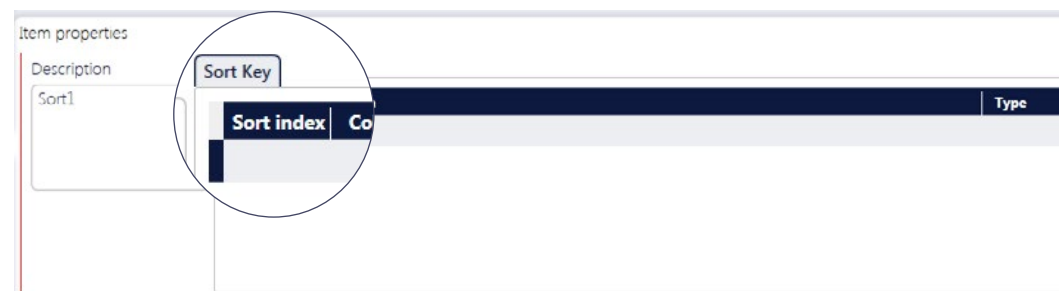
Operator SORT umożliwia zmianę kolejności rekordów w strumieniu danych, znajdujących się na magistrali wejściowej. Na magistrali wyjściowej znajduje się strumień w sekwencji, zgodnej z warunkami zadeklarowanymi w operatorze SORT.



Przykład projektu GRAVITY z użyciem operatora **SORT**.

### KONFIGURACJA KLUCZA SORTUJĄCEGO

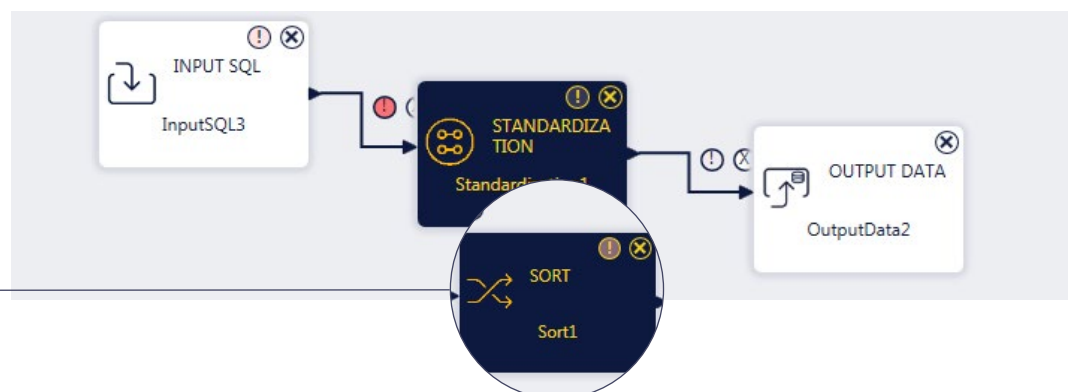
Możesz ustawić klucz sortujący składający się dowolnej liczby kolumn magistrali. Dla każdej kolumny określasz typ, czyli kierunek sortowania (od największej do najmniejszej lub odwrotnie).



### 8.3.14 STANDARIZATION

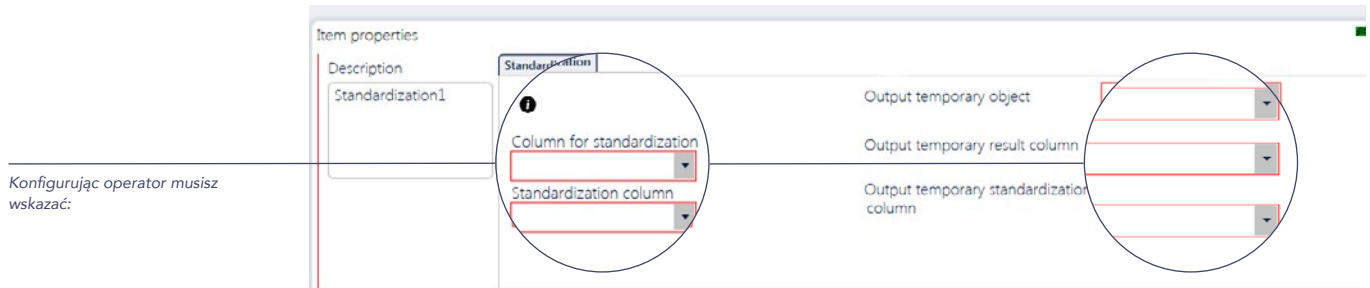


Operator STANDARIZATION umożliwia zmianę wartości danej na podstawie identyfikatora uniwersalnego, zawartego w jednej z kolumn strumienia wejściowego. Nowa wartość jest wyszukiwana na podstawie identyfikatora uniwersalnego w słowniku pobranym niezależnie.



Przykład projektu GRAVITY z użyciem operatora **STANDARIZATION**.

## KONFIGURACJA OPERATORA



Konfigurując operator musisz wskazać:

- Kolumnę z magistrali wejściowej, podlegającą standaryzacji (**COLUMN FOR STANDARIZATION**) (ta kolumna będzie przekształcana)
- Kolumnę standaryzującą (**STANDARIZATION COLUMN**) (kolumna zawierająca identyfikator uniwersalny)
- Dane zewnętrzne, na podstawie których dokonamy konwersji (dane są zawarte w operatorze **OUTPUT TEMPORARY** jako wyniku innego strumienia przetwarzania)
- Kolumnę z **OUTPUT TEMPORARY (OUTPUT TEMPORARY STANDARIZATION COLUMN)** (kolumna zawierająca identyfikator uniwersalny; łączenie z danymi strumienia wejściowego magistrali operatora)
- Kolumnę z **OUTPUT TEMPORARY (OUTPUT TEMPORARY RESULT COLUMN)** (kolumna wynikowa)

**Standardization**

An example of standardization integration of data from different sources.  
Standardize contractorId based on common nip:

Column For Standardization = column with value to change [ContractorId].  
Standardization Column = column with common value [ContractorNip].  
Output Temporary Standardization Column = column in OT with common value [Nip].  
Output Temporary Result Column = column in OT with new value column for standardization [Id].

In Data			Output temporary (dictionary with contractors)			Out Data		
Index	ContractorId	ContractorNip	Id	Nip	Name	Index	ContractorId	ContractorNip
1	176	1331976787	10	1331976787	Company10	1	10	1331976787
2	161	6138888995	11	9391929531	Company11	2	161	6138888995
5	763	5653146579	12	5653146579	Company12	5	12	5653146579

Przykład akcji standaryzacyjnej.

### 8.3.15 STANDARIZATION OUTPUT



Operator STANDARIZATION OUTPUT przeprowadza akcję, która jest szczególnym przypadkiem standaryzacji.

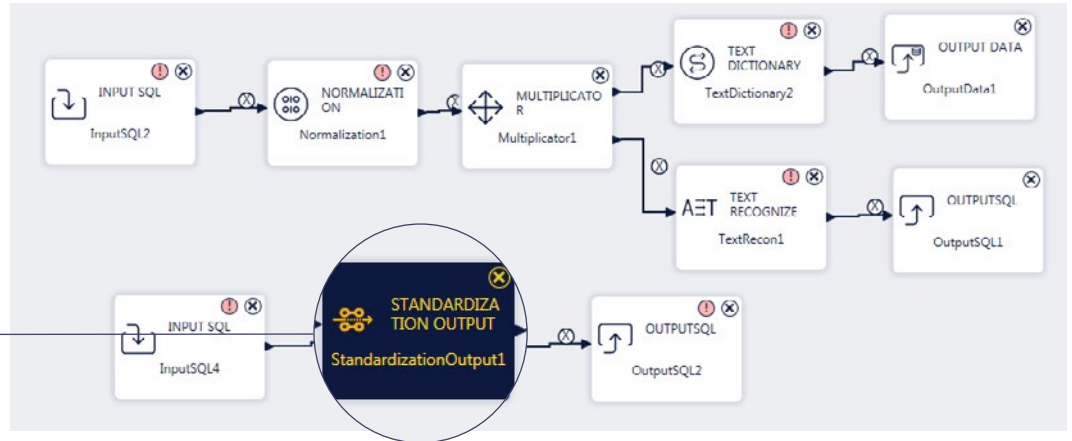
Kontekst

Zwykły operator STANDARIZATION umożliwia zmianę wartości danej, na podstawie identyfikatora uniwersalnego zawartego w jednej z kolumn strumienia wejściowego. Nowa wartość jest wyszukiwana na podstawie identyfikatora uniwersalnego w słowniku pobranym niezależnie.

Definicja STANDARIZATION OUTPUT

Operator STANDARIZATION OUTPUT umożliwia zamianę wartości danej (kolumny), na podstawie słownika tworzonego automatycznie ad hoc przez operator OUTPUT SQL podczas zapisywania strumienia danych w bazie. Istnieje możliwość takiego skonfigurowania OUTPUT SQL, aby niektóre dane (kolumny) zmieniały wartość (tak np. dzieje się z identyfikatorami unikalnymi obiektów uniwersalnych w przypadku, gdy dane pochodzą z wielu źródeł). Zmiany są zapamiętywane w tabeli przekodowań, utworzonej w bazie docelowej przez GRAVITY i dostępne, poprzez operator STANDARIZATION OUTPUT. Innymi słowy, możesz sobie wyobrazić, że jest to standaryzacja, nie na podstawie słownika zewnętrznego OUTPUT TEMPORARY, lecz na podstawie słownika przekodowań tworzonych przez operator OUTPUT SQL.

Przykład projektu GRAVITY z użyciem operatora **STANDARDIZATION OUTPUT**.



### KONFIGURACJA OPERATORA

Item properties

Description: StandardizationOutput1

Output sql object	Output sql pk mapping
Output sql object: OutputsSQL1	
Table name: ap.gm_jednostki	
Action for error: Null	
	Null
	Stop with error

W pierwszym kroku (patrz ilustracja powyżej), musisz wskazać operator **OUTPUT SQL** (w aplikacji GRAVITY może być wiele takich operatorów), który jest odpowiedzialny za tworzenie słownika przekodowań, tabelę docelową, podlegającą standaryzacji podczas akcji w **OUTPUT SQL** (GRAVITY podpowie Tobie tylko te tabele, które biorą udział w akcji wybranego **OUTPUT SQL**) oraz zadeklarować zachowanie GRAVITY w przypadku wykrycia błędu (np. brak pozycji w słowniku przekodowań) podczas standaryzacji. Możesz zdecydować się na akcję zatrzymaj działanie GRAVITY lub idź dalej, a jako wartość danej przetwarzanej podstaw **NULL**.

Output sql object | Output sql pk mapping

Output pk column name	In column
id_kontrahenta	(1) id_kontrahenta

W zakładce **OUTPUT SQL PK** musisz powiązać kolumnę z magistrali wejściowej z identyczną kolumną słownika stworzonego przez operator **OUTPUT SQL**.

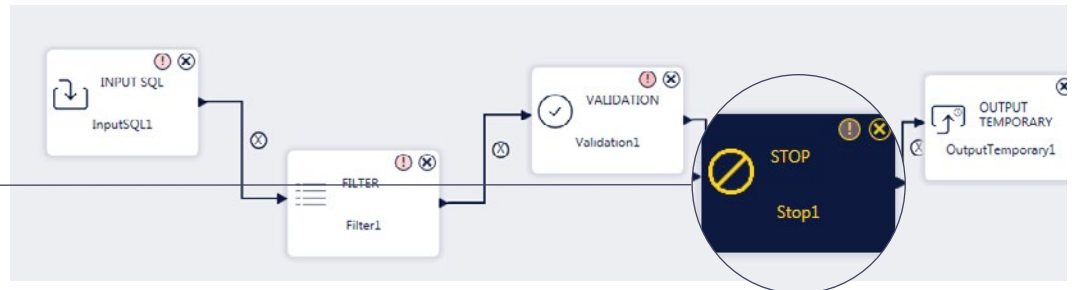


## 8.3.16 STOP



Operator STOP umożliwia zatrzymanie całego procesu przetwarzania danych w przypadku spełnienia zadeklarowanego warunku.

Przykład projektu GRAVITY z użyciem operatora **STOP**.



## KONFIGURACJA OPERATORA



Deklarując warunek zatrzymania (patrz ilustracja powyżej), masz do dyspozycji słownik funkcji (patrz ilustracja poniżej).

DisplayName	Category	Description
GET_AGGR	methods	Aggregation values of incoming column
GET_OUTTEMP_AGGR	methods	Aggregation values of output temporary column
id_kontrahenta	columns	Value of column 'id_kontrahenta'
skrot nazwy	columns	Value of column 'skrot nazwy'

Dostępne funkcje agregujące:

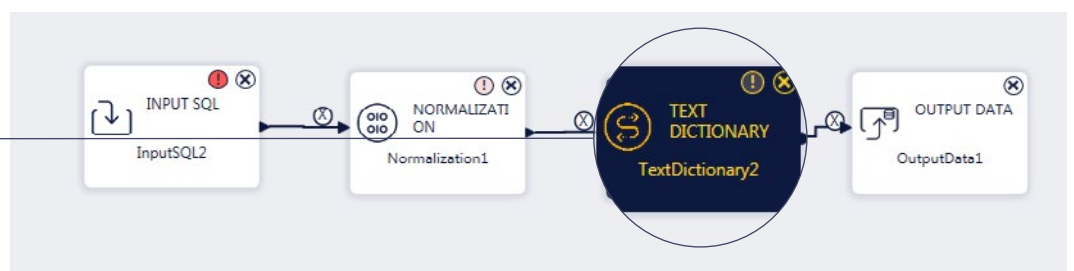
- **GET\_AGGR** ...funkcja sumująca wartość wskazanej kolumny w strumieniu danych
- **GET\_OUTTEMP\_AGGR** ...funkcja sumująca wartość wskazanej kolumny w zbiorze typu **OUTPUT TEMPORARY**

## 8.3.17 TEXT DICTIONARY



Operator TEXT DICTIONARY zamienia wartość wybranej kolumny na nową wartość, zgodnie ze stworzonym słownikiem.

Przykład projektu GRAVITY z użyciem operatora **TEXT DICTIONARY**.



## KONFIGURACJA OPERATORA

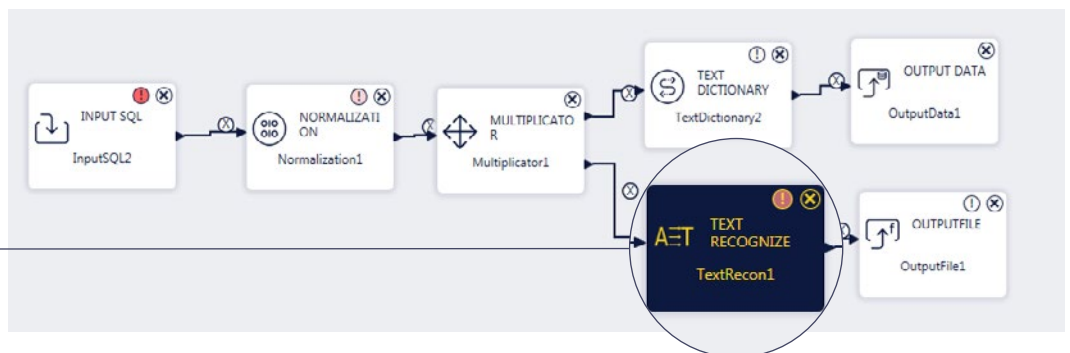
Słownik przekodowań tworzysz wskazując kolumnę, wartość dotychczasową i nową wartość. Słownik może mieć dowolną liczbę przekodowań.



### 8.3.18 TEXT RECOGNIZE

Operator TEXT RECOGNIZE przetwarza dane w ten sposób, że porównuje wartość danej do uniwersalnego słownika. W przypadku, gdy nie wykryje danej w słowniku, znajduje wartość zamieszczoną w słowniku, najbardziej prawdopodobną do wartości badanej, a następnie zamienia wartość dotychczasową na wartość słownikową. Słownik uniwersalny może powstać w dwojaki sposób: może zostać wskazany przez użytkownika (jednoznacznie zadeklarowany) lub powstać automatycznie na podstawie badania skali powtórzeń.

Przykład projektu GRAVITY z użyciem operatora TEXT RECOGNIZE.



## KONFIGURACJA OPERATORA

Podczas konfiguracji, wskazujesz kolumnę podlegającą rozpoznaniu oraz wariant akcji rozpoznania.



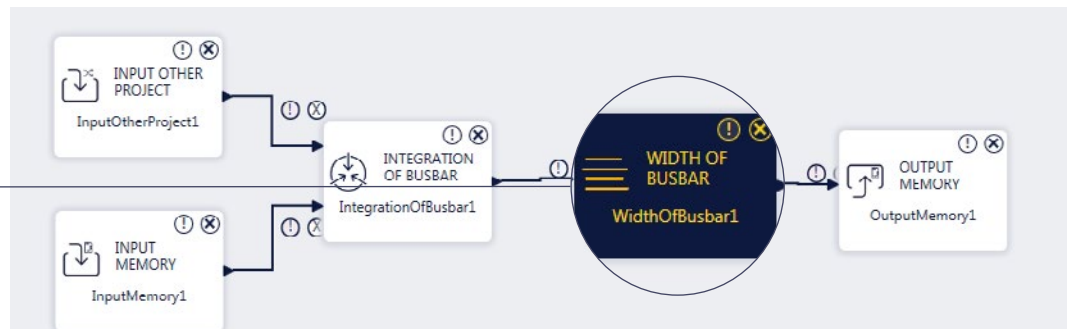
Jeżeli wybierzesz wariant rozpoznania na podstawie stałego słownika, musisz dodatkowo wskazać zbiór **OUTPUT TEMPORARY**, który zawiera słownik uniwersalny oraz kolumnę pozostającą w relacji do kolumny badanej podczas akcji.

### 8.3.19 WIDTH OF BUSBAR



Operator WIDTH OF BUSBAR pozwala na zmianę szerokości magistrali. Magistrala wyjściowa – jej szerokość oraz nazwy kolumn – jest deklarowana podczas konfiguracji, na podstawie kolumn magistrali wejściowej.

Przykład projektu GRAVITY z użyciem operatora **WIDTH OF BUSBAR**.



### KONFIGURACJA OPERATORA

Konfigurując, będziesz mógł zadeklarować każdy element szerokości magistrali wyjściowej (czyli każdą kolumnę) na podstawie kolumn magistrali wejściowej.

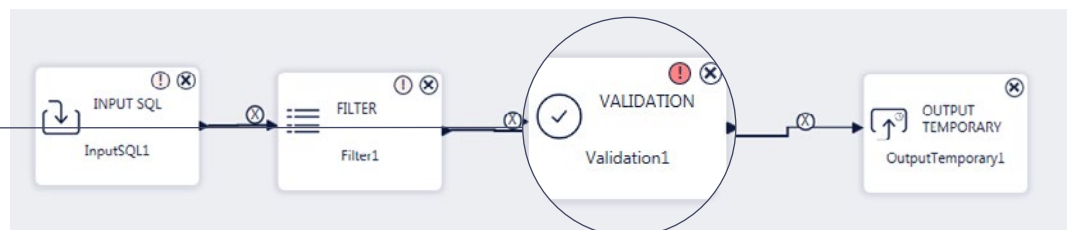


### 8.3.20 VALIDATION



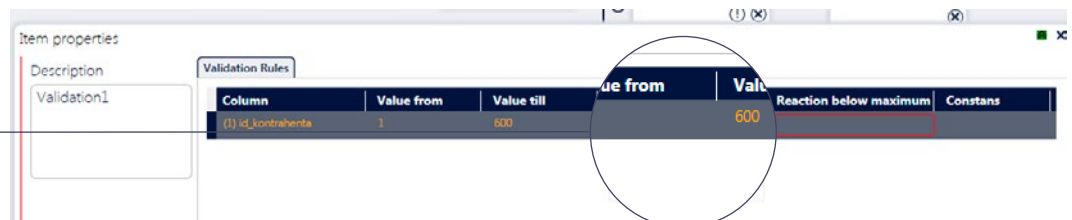
Operator VALIDATION bada wartości zadeklarowanych danych (kolumn) strumienia znajdującego się na magistrali wejściowej. W przypadku wykrycia wartości nie należących do zadeklarowanych (walidowanych) przedziałów, system wykonuje jedną z akcji: nadaje nową wartość na podstawie limitu (min lub max w zależności od miejsca przekroczenia), wartość constans (zadeklarowaną) lub usuwa rekord ze strumienia danych.

Przykład projektu GRAVITY z użyciem operatora **VALIDATION**.



### KONFIGURACJA OPERATORA

Podczas konfiguracji, wskazujesz kolumnę podlegającą rozpoznaniu oraz wariant akcji rozpoznania.



Musisz wskazać kolumnę magistrali wejściowej (patrz ilustracja powyżej), która podlega walidacji, przedział wartości, który jest prawidłowy oraz rodzaj akcji w przypadku przekroczenia przedziałów. Możesz zadeklarować niezależne akcje dla przypadku przekroczenia minimum oraz maksimum. Możesz zdecydować się na następujące akcje:

Nadać nową wartość równą minimum (jeżeli wartość jest poniżej minimum)

- Nadać nową wartość równą maksimum (jeżeli wartość jest powyżej maksimum)
- Nadać wartość constans jeżeli wartość jest poza przedziałem walidacji
- Filtrować rekord jeżeli wartość jest poza przedziałem walidacji

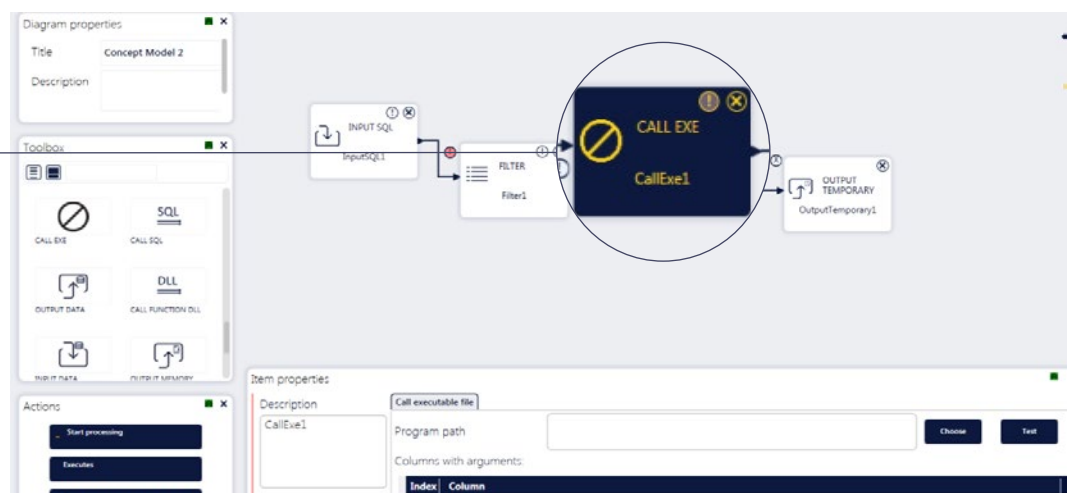
## 8.4 OPERATORY PRZETWARZANIA ZEWNĘTRZNEGO

### 8.4.1 CALL EXE



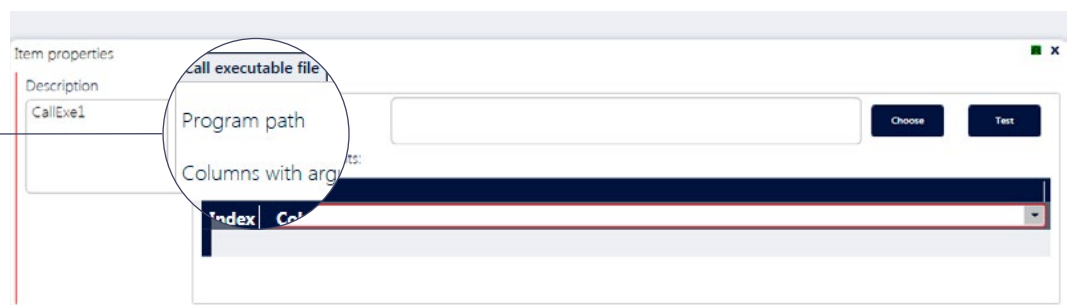
Operator CALL EXE umożliwia wywołanie zewnętrznego programu EXE, dla każdego rekordu znajdującego się na magistrali wejściowej do operatora. Możliwe wywołanie programu zewnętrznego z parametrami znajdującymi się na magistrali danych.

Przykład projektu GRAVITY z użyciem operatora CALL EXE.



#### USTAWIENIE PARAMETRU: PROGRAM PATH

W opcji ustawiasz ścieżkę dostępu do programu zewnętrznego.



#### USTAWIENIE ARGUMENTU WYWOŁANIA PROGRAMU ZEWNĘTRZNEGO

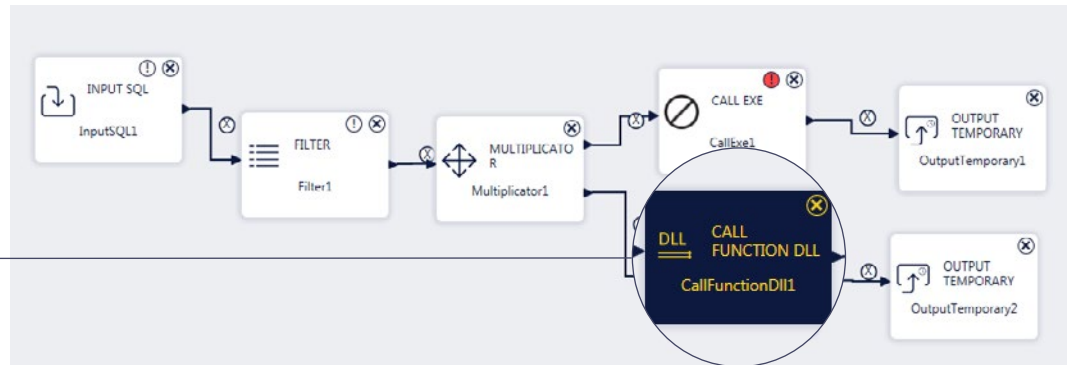
W opcji, ustawiasz potencjalne argumenty wywołania programu zewnętrznego. Jako parametr, wskazujemy kolumnę magistrali na wejściu do operatora.



## 8.4.2 CALL FUNCTION DLL



Operator CALL FUNCTION DLL umożliwia wywołanie zewnętrznej funkcji umieszczonej w bibliotece DLL, dla każdego rekordu znajdującego się na magistrali wejściowej do operatora. Możliwe wywołanie funkcji zewnętrznej z parametrami znajdującymi się na magistrali danych.

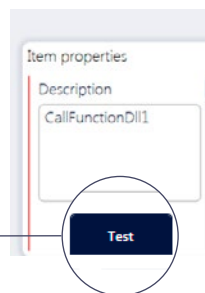


Przykład projektu GRAVITY z użyciem operatora **CALL DLL**.

### USTAWIENIE PARAMETRU: PROGRAM PATH



W opcji, ustawiasz ścieżkę dostępu do funkcji zewnętrznej oraz deklarujesz nazwę funkcji.



W opcji powyżej, masz możliwość przetestowania wywołania zadeklarowanej funkcji zewnętrznej.

### WSKAZANIE MIEJSCA W MAGISTRALI JAKO NOŚNIKA REZULTATU

W opcji wskazujesz, która kolumna magistrali będzie nośnikiem rezultatu działania funkcji zewnętrznej.

Aby właściwie użyć funkcji zewnętrznej, musisz znać typ danych zwracanych przez funkcję zewnętrzną. Edytując pole, automatycznie podpowie się słownik z wariantami dostępnych typów danych.

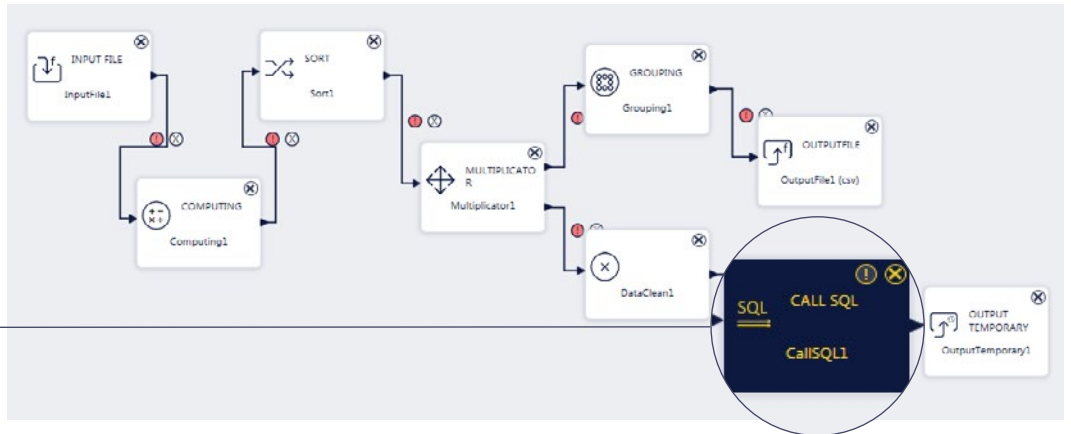
### USTAWIENIE ARGUMENTU WYWOŁANIA FUNKCJI ZEWNĘTRZNEJ

W opcji, ustawiasz potencjalne argumenty wywołania funkcji zewnętrznej. Jako parametr wskazujemy kolumnę magistrali na wejściu do operatora.

### 8.4.3 CALL SQL

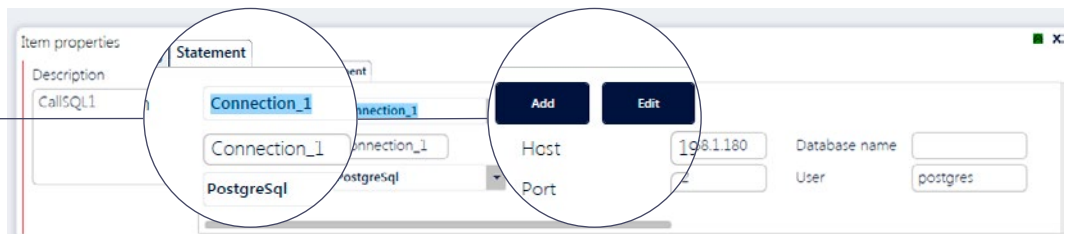


Operator CALL SQL wykonuje akcję SQL na zadeklarowanej bazie SQL z wykorzystaniem danych znajdujących się na magistrali wejściowej.



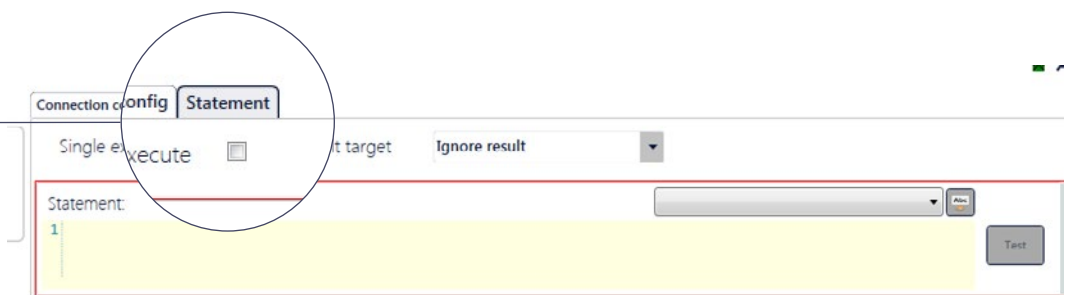
Przykład projektu GRAVITY z użyciem operatora **CALL SQL**.

### KONFIGURACJA OPERATORA



Podłączenie możesz wskazać ze słownika, jeżeli wcześniej definiowałeś takie podłączenia, lub zdefiniować nowe (opcje **ADD** lub **EDIT**).

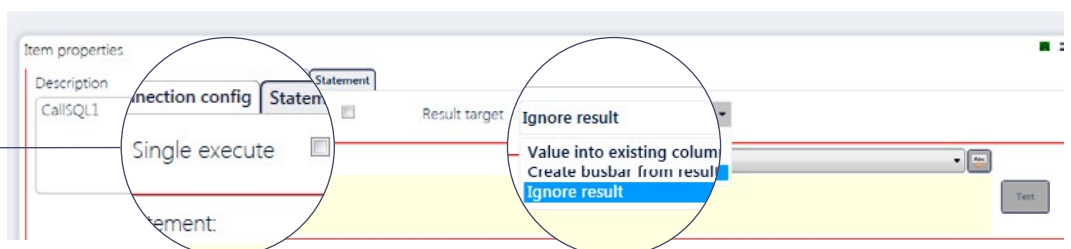
Operator służy do wykonania kodu **SQL** w bazie danych, a zatem przede wszystkim musisz wskazać podłączenie do wybranej bazy.



Możesz dokonać wyboru bez kłopotliwej edycji

W zakładce **STATEMENT** możesz edytować wyrażenie **SQL**. Dostęp do parametrów (w tym wszystkich kolumn magistrali wejściowej) zapewnia słownik.

Całe wyrażenie można przetestować wybierając opcję **TEST**.



Możesz jednak zdecydować się na wariant jednokrotnego wywołania: należy wówczas zaznaczyć opcję **SINGLE EXECUTE**.

Masz dwa warianty wykorzystania operatora. Wariant pierwszego wyboru, to wariant wywoływania kodu **SQL** dla każdego rekordu, znajdującego się w strumieniu danych na magistrali wejściowej.

Kod **SQL** może zwracać wynik, który jest przetworzony w różny sposób, w zależności od nastawy **RESULT TARGET** (patrz ilustracja powyżej):

- **IGNORE RESULT**.....rezultat kodu **SQL** nie ma żadnego wpływu; na wyjściu magistrali pojawi się strumień danych identyczny jak na magistrali wejściowej;
- **VALUE INTO EXISTING COLUMN**.....rezultat kodu **SQL** będzie umieszczony we wskazanej kolumnie magistrali wyjściowej (pozostałe dane magistrali wejściowej zachowują swoje wartości);
- **CREATE BUSBAR FROM RESULT**.....magistrala wyjściowa jest zgodna z wynikiem wygenerowanym przez kod **SQL** (ma zastosowanie dla **SINGLE EXECUTE**);

#### 8.4.4 CALL WEB SERVICE

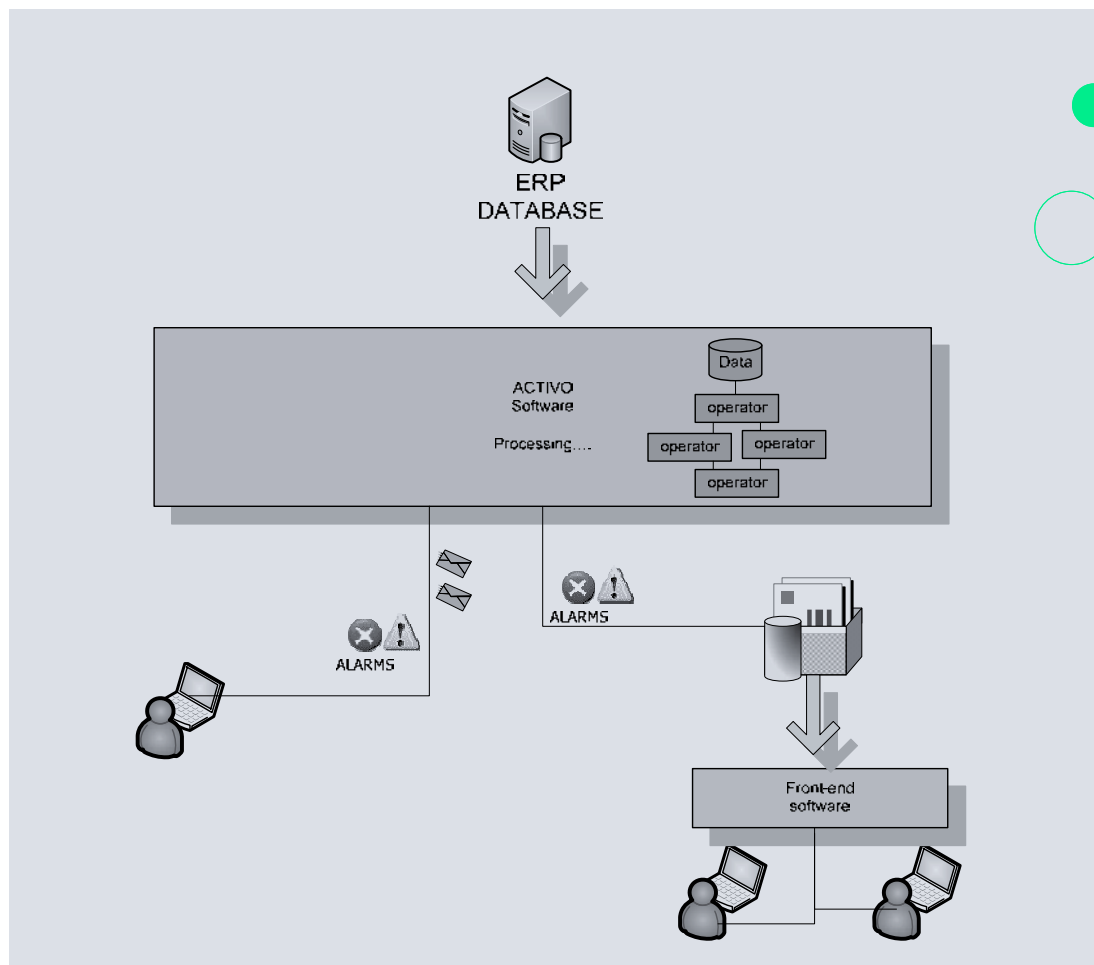


Operator **CALL WEB SERVICE** wywołuje zewnętrzną funkcję **WEB SERVICE**. W takim wypadku wywoływana zewnętrzna funkcja **WEB SERVICE** pełni zadanie zewnętrznego operatora i danych wejściowych do operatora **CALL WEB SERVICE**. Konfigurując operator deklarujesz strukturę protokołu **JSON** zasilającego zewnętrzną funkcję **WEB SERVICE** (deklaracja jest oczywiście w pełni zautomatyzowana: deklarujesz na podstawie pliku wzorcowego **JSON** i sygnału magistrali wejściowej). Ponadto konfigurujesz strukturę strumienia danych rezultatu przetwarzania, przy założeniu, że jest to protokół **JSON**. Deklaracja jest automatyczna, na podstawie pliku wzorcowego oraz sygnału magistrali wyjściowej. **GRAVITY** obsługuje serwery **WEB SERVICE** z protokołem **REST**.

## 9. PRZYKŁADY ZASTOSOWAŃ

Poniżej przytaczamy przykłady zastosowań GRAVITY. To tylko garść inwencji. Pomysłów zastosowania może być nieskończenie wiele.

### 9.1 ALARMY WEWNĘTRZNE



#### KONTEKST

Przedsiębiorstwo użytkujące oprogramowanie klasy ERP, chce stworzyć system automatycznego powiadamiania swoich pracowników. System powiadamiania ma być automatyczny i niezależny od użytkownika. System powiadamiania ma automatycznie wykrywać sytuacje alarmowe i powiadamiać właściwych adresatów.

#### ZADANIE

Stworzenie funkcjonalności, która pomoże użytkownikowi natychmiast reagować na spadek sprzedaży.



### **PRZYKŁADOWE ZASTOSOWANIE GRAVITY**

Na użytek naszego przykładu zakładamy, że musimy informować o spadku sprzedaży z poprzednich 7 dni pod warunkiem, że wartość sprzedaży 7-dniowej była mniejsza, niż mediana za okres ostatnich 12 – miesięcy. W przypadku rozpoznania warunku, który sobie założyliśmy GRAVITY wysyła informację pocztą elektroniczną do skonfigurowanego adresata.

Impuls wywołujący przetwarzanie: skanowanie wywołań czasowych.

### **ZADANIE**

Stworzenie funkcjonalności, która pomoże użytkownikowi natychmiast pozyskiwać informację o przekroczeniu kredytu kupieckiego przez kontrahenta.

### **PRZYKŁADOWE ZASTOSOWANIE GRAVITY**

W przypadku rozpoznania warunku – przekroczenie kredytu kupieckiego – GRAVITY wysyła informację pocztą elektroniczną do skonfigurowanego adresata (adresatów).

Impuls wywołujący przetwarzanie: skanowanie wywołań czasowych.

### **PRZYKŁADOWE ZASTOSOWANIE GRAVITY**

Jak wyżej z jedną bardzo ważną różnicą. Impuls wywołujący przetwarzanie to notyfikacja bazy danych. Momentem wyzwolenia notyfikacji jest dodanie rekordu faktury dla kontrahenta.

### **PRZYKŁADOWE ZASTOSOWANIE GRAVITY**

Identycznie, jak w zastosowaniu powyżej z tą różnicą, że informacja o wykrytym zdarzeniu jest wysyłana jako SMS do wybranej osoby (zastosowanie operatora [CALL WEB SERVICE](#) jako funkcji zewnętrznej obsługującej bramkę SMS).

### **ZADANIE**

Stworzenie funkcjonalności, która pomoże użytkownikowi natychmiast pozyskiwać informację o przeterminowaniu realizacji zamówień klienckich.

### **PRZYKŁADOWE ZASTOSOWANIE GRAVITY**

W przypadku rozpoznania warunku (istnienie przeterminowań), GRAVITY wysyła informację pocztą elektroniczną do skonfigurowanego adresata (adresatów).

Impuls wywołujący przetwarzanie: skanowanie wywołań czasowych.

### **PRZYKŁADOWE ZASTOSOWANIE GRAVITY**

W przypadku rozpoznania warunku (istnienie przeterminowań niezamkniętych zamówień klienta), GRAVITY wysyła pozostawia informację o wszystkich przypadkach przeterminowań na magistralę wiodącą, do bazy tymczasowej (sesyjny cache pamięciowy).

Ponadto skonfigurowano funkcję [WEB SERVICE](#) dającą dostęp do bazy pamięciowej.

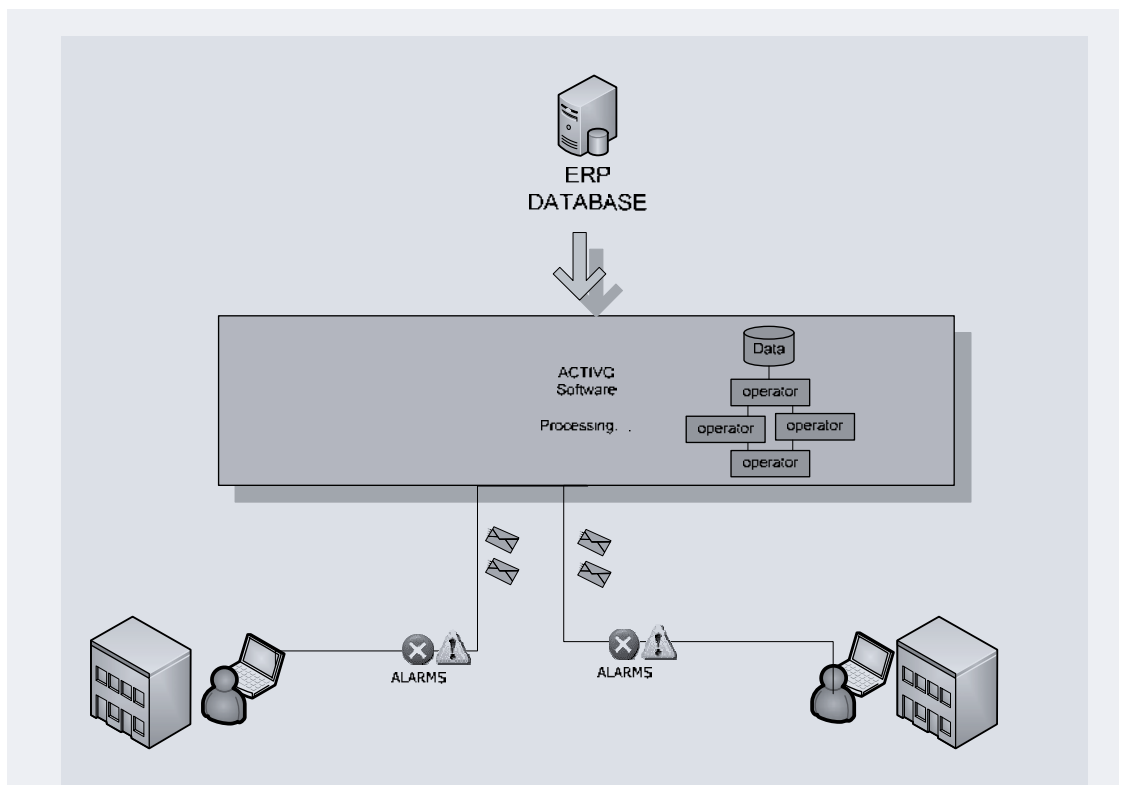
Dzięki temu, każde oprogramowanie zewnętrzne ma dostęp do przygotowanej informacji rezydującej w pamięci.

Impuls wywołujący przetwarzanie: skanowanie wywołań czasowych.



Uwaga: jeżeli użytkownik chce mieć zawsze świeżą informację w bazie pamięciowej (do której mamy akces poprzez funkcję **WEB SERVICE**) może ustawić interwał przerwania czasowych odświeżających bazę (czyli wywołujących niniejszy projekt) na bardzo krótki czas.

## 9.2 ALARMY ZEWNĘTRZNE



### KONTEKST

Przedsiębiorstwo użytkujące oprogramowanie klasy ERP, chce stworzyć system automatycznego powiadamiania swoich kontrahentów. System powiadamiania ma być automatyczny i niezależny od użytkownika. System powiadamiania ma automatycznie wykrywać sytuacje alarmowe i powiadamiać właściwych adresatów.

### ZADANIE

Uruchomienie automatycznego powiadamiania kontrahentów, w przypadku wykrycia niezapłaconych i przeterminowanych faktur.

### PRZYKŁADOWE ZASTOSOWANIE GRAVITY

W przypadku rozpoznania warunku (istnienie przeterminowań zapłat na fakturach kontrahenta), GRAVITY wysła powiadomienia pocztą elektroniczną do adresatów.

Impuls wywołujący przetwarzanie: skanowanie wywołań czasowych.

Diagram przedstawia projekt zamodelowany w GRAVITY:

### ZADANIE

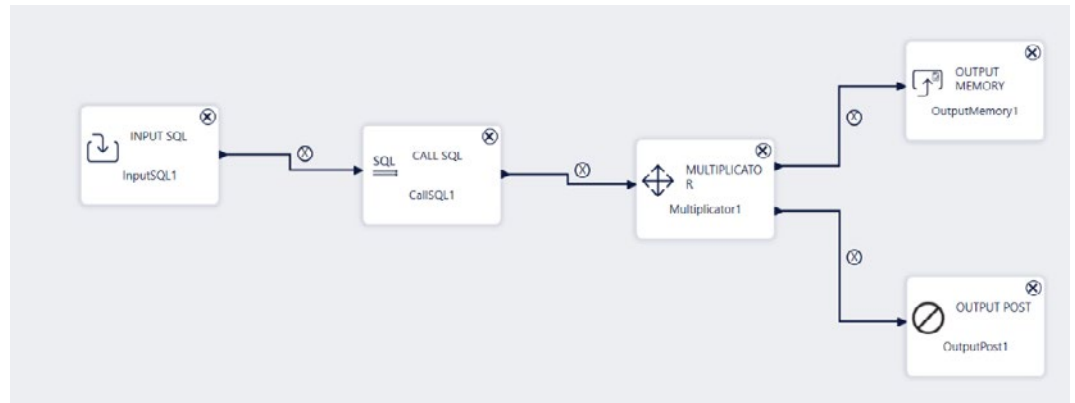
Uruchomienie automatycznego powiadamiania kontrahentów o aktualnych saldach z kwartalną częstotliwością.

### PRZYKŁADOWE ZASTOSOWANIE GRAVITY

Proces obliczania sald dla każdego kontrahenta, przesłania raportu oraz ustawienia znacznika w bazie źródłowej (użycie operatora **CALL SQL**).

Impuls wywołujący przetwarzanie: skanowanie wywołań czasowych.

Diagram przykładowy



### ZADANIE

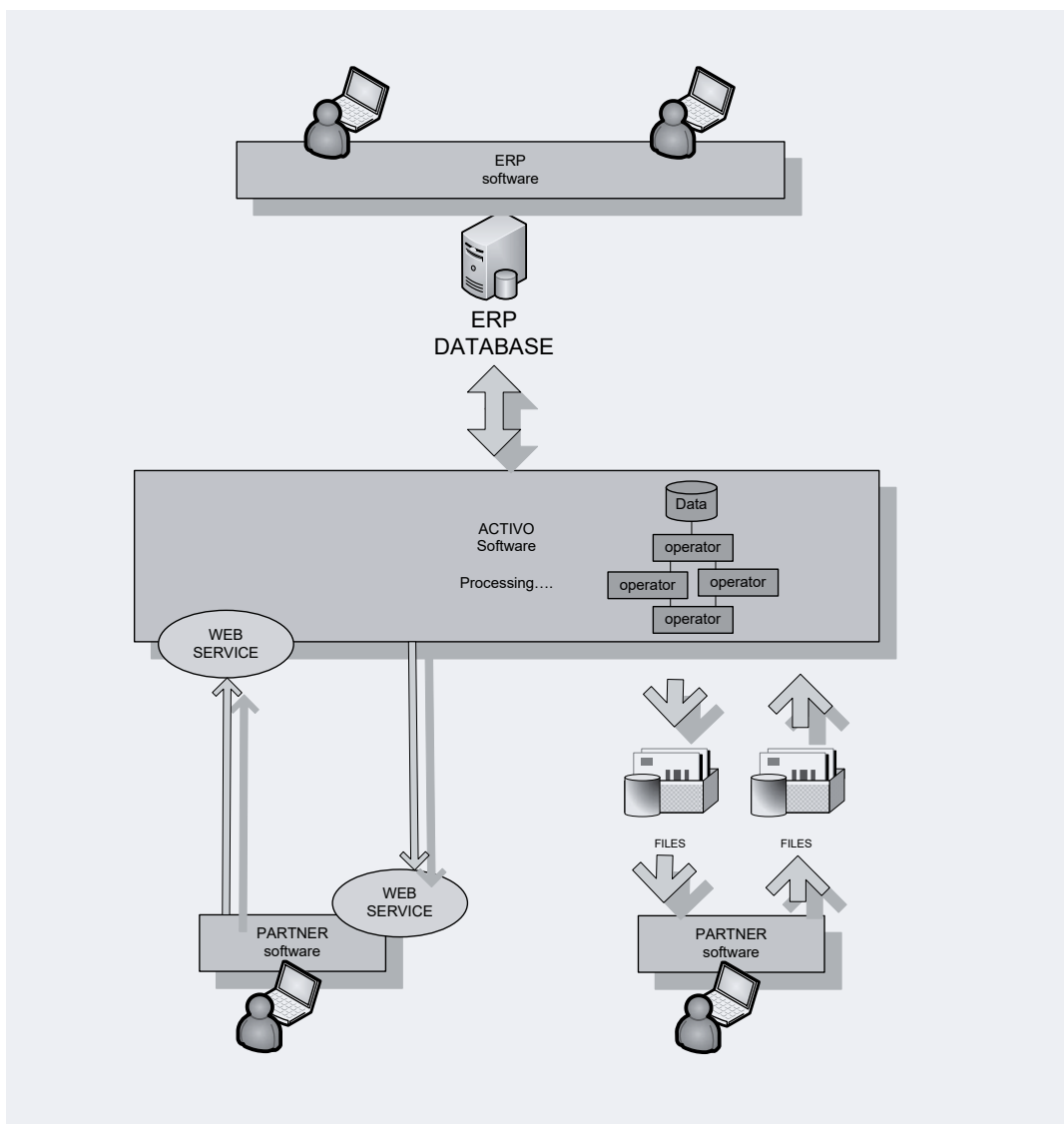
Uruchomienie automatycznego powiadomienia dostawcy w przypadku wykrycia przeterminowanej dostawy.

### PRZYKŁADOWE ZASTOSOWANIE GRAVITY

W przypadku rozpoznania warunku (istnienie przeterminowań w otwartych zamówieniach dostawy), GRAVITY wysła informację do dostawcy pocztą elektroniczną.

Impuls wywołujący przetwarzanie: skanowanie wywołań czasowych.

### 9.3 INTEGRACJA Z SYSTEMAMI INFORMATYCZNYMI ODBIORCÓW



#### **KONTEKST**

Przedsiębiorstwo współpracuje z odbiorcami, posiadającymi otwarte systemy informatyczne.

#### **ZADANIE**

Stworzenie funkcjonalności, które pomogą użytkownikowi zintegrować proces zamawiania produktów z systemami informatycznymi wybranych odbiorców.

### **PRZYKŁADOWE ZASTOSOWANIE GRAVITY**

Proces automatycznego pobrania zamówienia klienta. Zakładamy, że system informatyczny odbiorcy umieszcza plik z zamówieniem w określonej lokalizacji, specyficznej dla każdego kontrahenta. Lokalizacje są cały czas skanowana przez GRAVITY. W momencie wykrycia pliku zamówienia, jest ono dodawane przez GRAVITY do systemu ERP z wykorzystaniem funkcji WEB SERVICE systemu ERP (operator CALL WEB SERVICE). Powyższy przykład ma sens, tylko w kontekście ERP posiadającego funkcje WEB SERVICE.

Pobrany plik zamówienia zostaje umieszczony w innej lokalizacji (zamówienia pobrane).

### **PRZYKŁADOWE ZASTOSOWANIE GRAVITY**

Automatyczna akcja informowania odbiorcy o pobraniu automatycznym lub wprowadzeniu manualnym zamówienia. Akcja oparta na wywołaniu poprzez notyfikację bazy danych. Na adres mailowy kontrahenta, zostaje wysłana informacja o rozpoczęciu przetwarzania zamówienia (operator OUTPUT POST).

### **PRZYKŁADOWE ZASTOSOWANIE GRAVITY**

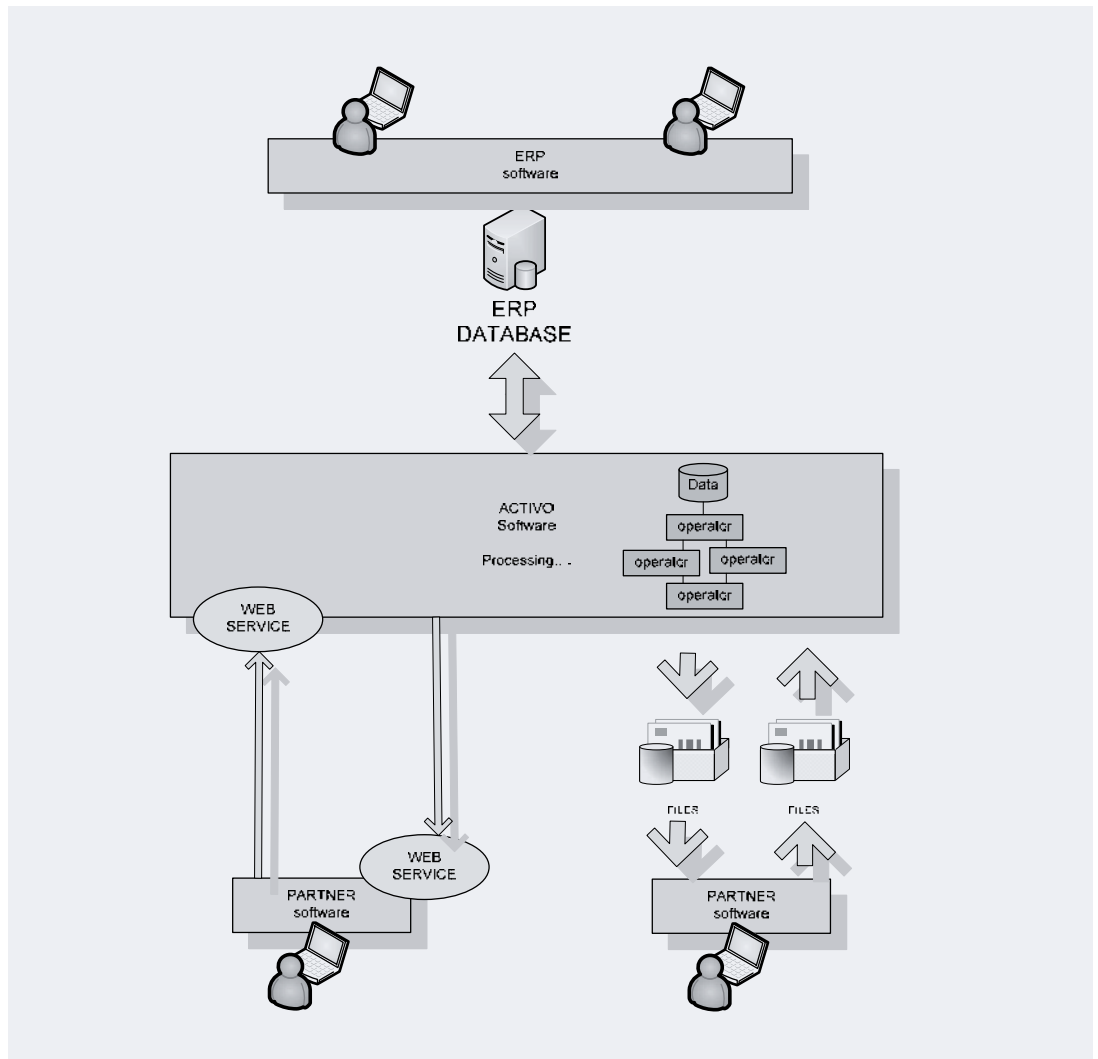
Automatyczna akcja informowania odbiorcy o zmianie statusu zamówienia (np. przesłane do realizacji). Akcja oparta na wywołaniu, poprzez notyfikację bazy danych. Na adres mailowy kontrahenta zostaje wysłana informacja o zmianie statusu zamówienia (operator OUTPUT POST). Oprócz informacji, wysyłany jest plik z pozycjami zamówienia klienta.



Uwaga: identyczne akcje mogłyby być zaimplementowane dla dowolnej liczby statusów zamówienia klienta.



## 9.4 INTEGRACJA Z SYSTEMAMI INFORMATYCZNYMI DOSTAWCÓW



### KONTEKST

Przedsiębiorstwo współpracuje z dostawcami posiadającymi otwarte systemy informatyczne.

### ZADANIE

Stworzenie funkcjonalności, które pomogą użytkownikowi zintegrować proces zamawiania dostaw materiałów z systemami informatycznymi wybranych dostawców.

### PRZYKŁADOWE ZASTOSOWANIE GRAVITY

Proces automatycznego wysłania zamówienia dostawy. Akcja oparta na wywołaniu poprzez notyfikację bazy danych (wykrycie zatwierdzenia zamówienia dostawy). Na adres mailowy dostawcy zostaje wysłane zamówienie dostawy (operator **OUTPUT POST**).

### **PRZYKŁADOWE ZASTOSOWANIE GRAVITY**

---

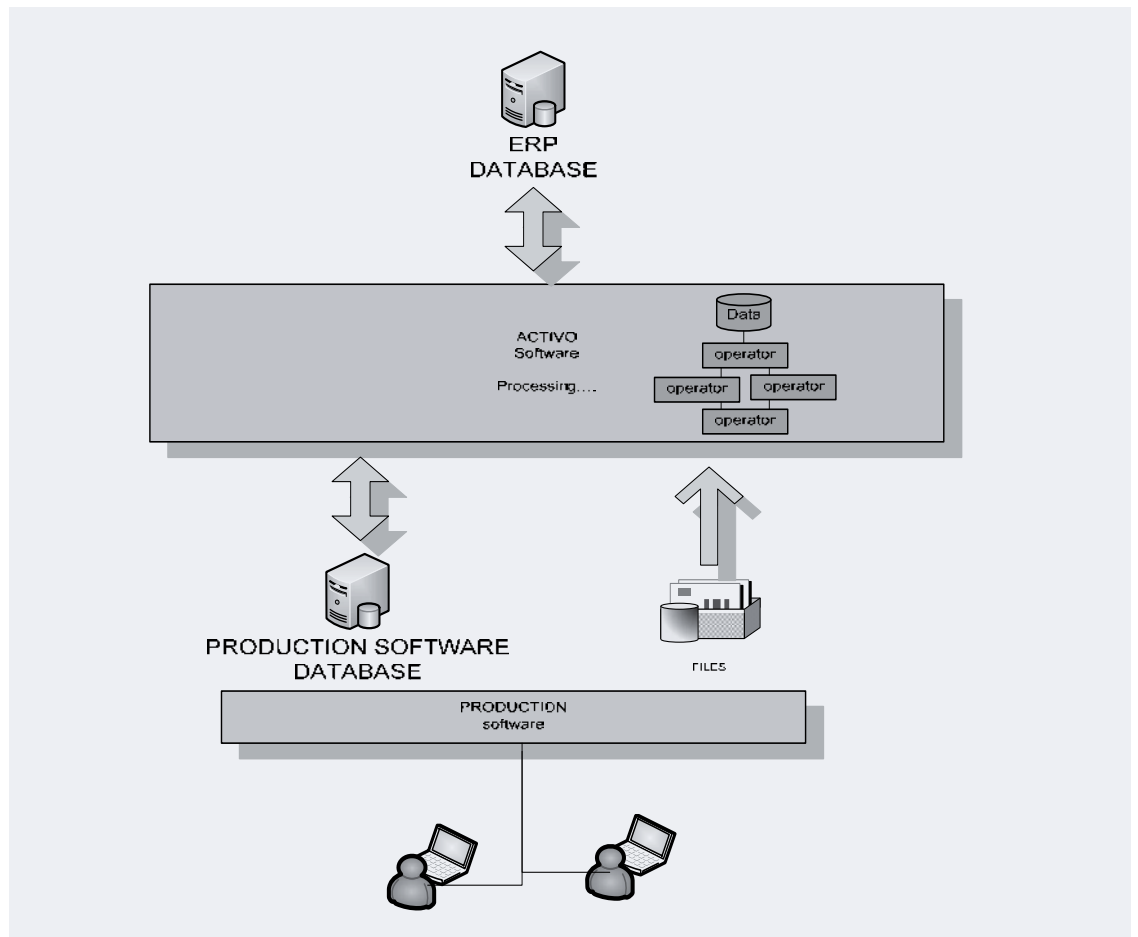
Automatyczne pobranie dokumentu przyjęcia dostawy.

Zakładamy, że system informatyczny dostawcy umieszcza plik z zamówieniem w określonej lokalizacji, specyficznej dla każdego dostawcy. Lokalizacje są cały czas skanowane przez GRAVITY. W momencie wykrycia pliku zamówienia dostawy, jest ono dodawane przez GRAVITY do systemu ERP, jako dokument magazynowy przyjęcia dostawy z wykorzystaniem funkcji WEB SERVICE systemu ERP (a więc z użyciem operatora CALL WEB SERVICE – zakładamy, że ERP posiada taką funkcję).

Pobrany plik zamówienia zostaje umieszczony w innej lokalizacji (zamówienia dostawy pobrane).



## 9.5 INTEGRACJA Z SYSTEMAMI WSPIERAJĄCYMI PRODUKCJĘ



### KONTEKST

Przedsiębiorstwo użytkuje oprogramowanie ERP oraz niezależne systemy dedykowane, wspierające produkcję.

### ZADANIE

Przedsiębiorstwo chce stworzyć warstwę pośrednią, integrującą systemy wspierające produkcję z oprogramowaniem ERP. Dzięki warstwie pośredniej, przedsiębiorstwo uniezależnia się od dostawców systemów integrowanych, oraz posiada znacznie większą zdolność wymiany systemów zintegrowanych.

### PRZYKŁADOWE ZASTOSOWANIE GRAVITY

Jednorazowa migracja danych: pobranie technologii zgromadzonych w plikach Excel. Dzięki stworzeniu projektu migracji, akcję można przeprowadzać wielokrotnie np. na użytek testów pilotażowych systemu informatyki wspierającego produkcję.

### **PRZYKŁADOWE ZASTOSOWANIE GRAVITY**

---

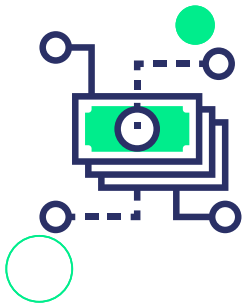
Integracja systemu ERP z dedykowanym oprogramowaniem dziedzinowym (np. typu CAD).

Transfer technologii oraz receptur materiałowych na życzenie użytkownika.

### **PRZYKŁADOWE ZASTOSOWANIE GRAVITY**

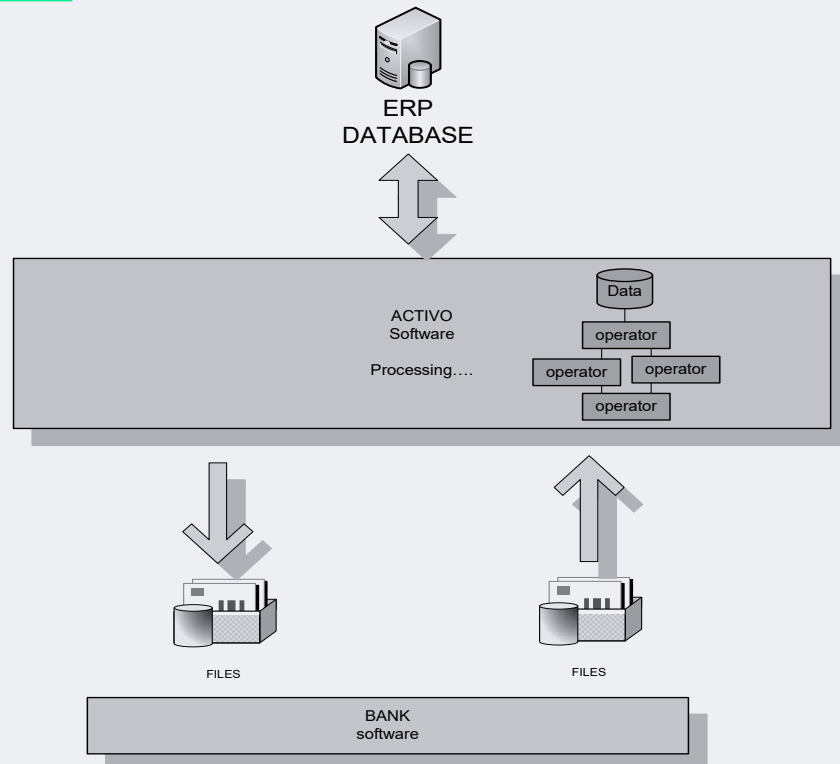
---

Śledzenie zaawansowania produkcji w toku. Integracja systemu ERP z dedykowanym oprogramowaniem obsługującym np. kartę wykonania operacji, detalu etc.



## 9.6 INTEGRACJA Z SYSTEMAMI BANKOWYMI

### KONTEKST



Przedsiębiorstwo użytkuje oprogramowanie **ERP** oraz stawia sobie za cel, zautomatyzowanie obsługi ewidencji bankowej.

### ZADANIE

Stworzenie funkcjonalności, które pomogą użytkownikowi zintegrować system **ERP** z bankiem.

### PRZYKŁADOWE ZASTOSOWANIE GRAVITY

Proces automatycznego pobierania wyciągów bankowych. Akcja oparta na wywołaniu poprzez terminarz czasowy. GRAVITY skanuje zadeklarowane lokalizacje, specyficzny dla każdego banku.

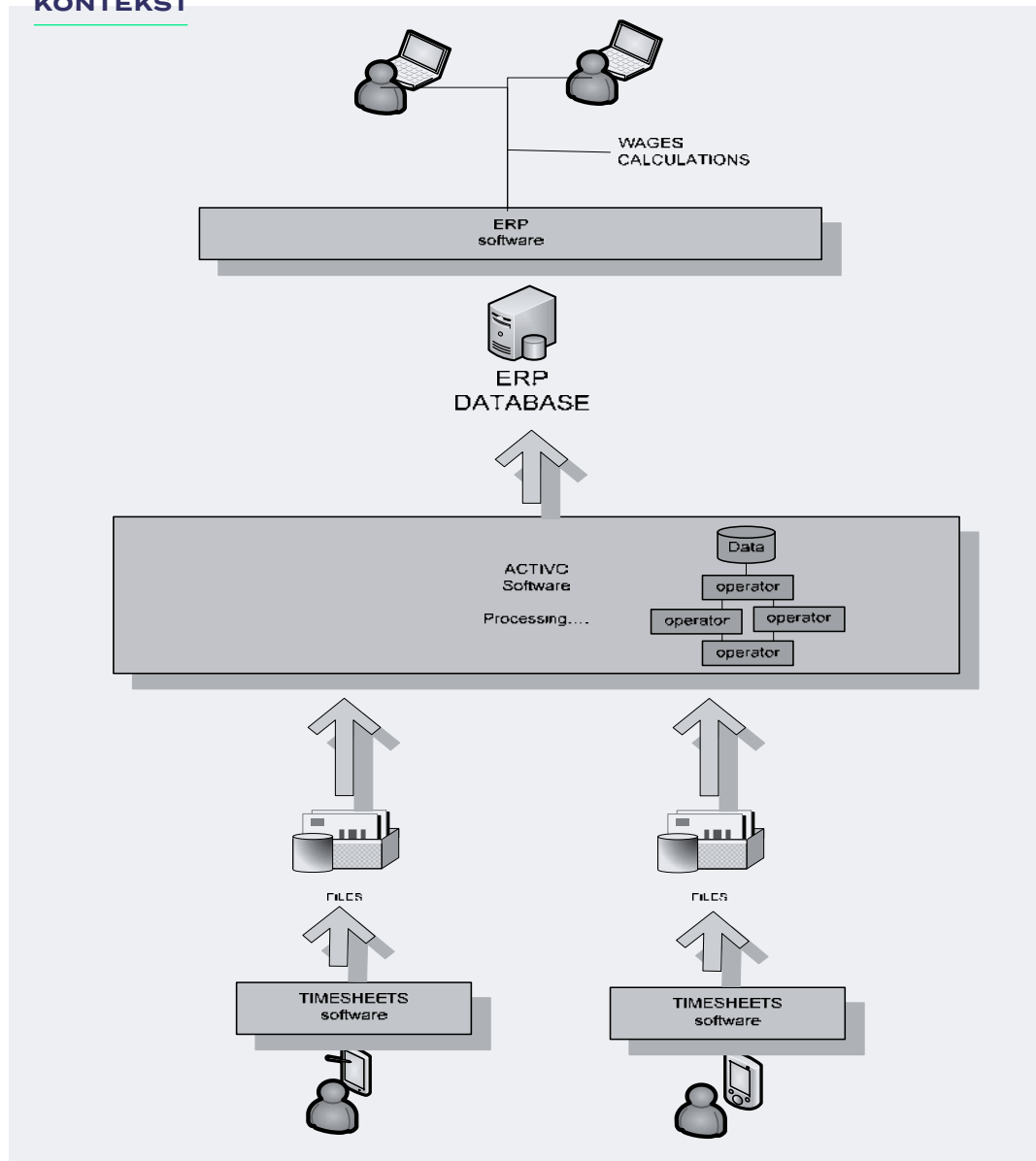
W momencie wykrycia pliku wyciągu bankowego, pozycje wyciągu są dodawane przez GRAVITY do systemu ERP z wykorzystaniem funkcji **WEB SERVICE** systemu **ERP**. Pobrany plik zamówienia zostaje umieszczony w innej lokalizacji (wyciągi bankowe pobrane).

## 9.7 INTEGRACJA



## EWIDENCJI CZASU PRACY Z OPROGRAMOWANIEM KADROWO – PŁACOWYM

### KONTEKST



Przedsiębiorstwo użykuje oprogramowanie ERP oraz system ewidencji czasu pracy od różnych dostawców.

### ZADANIE

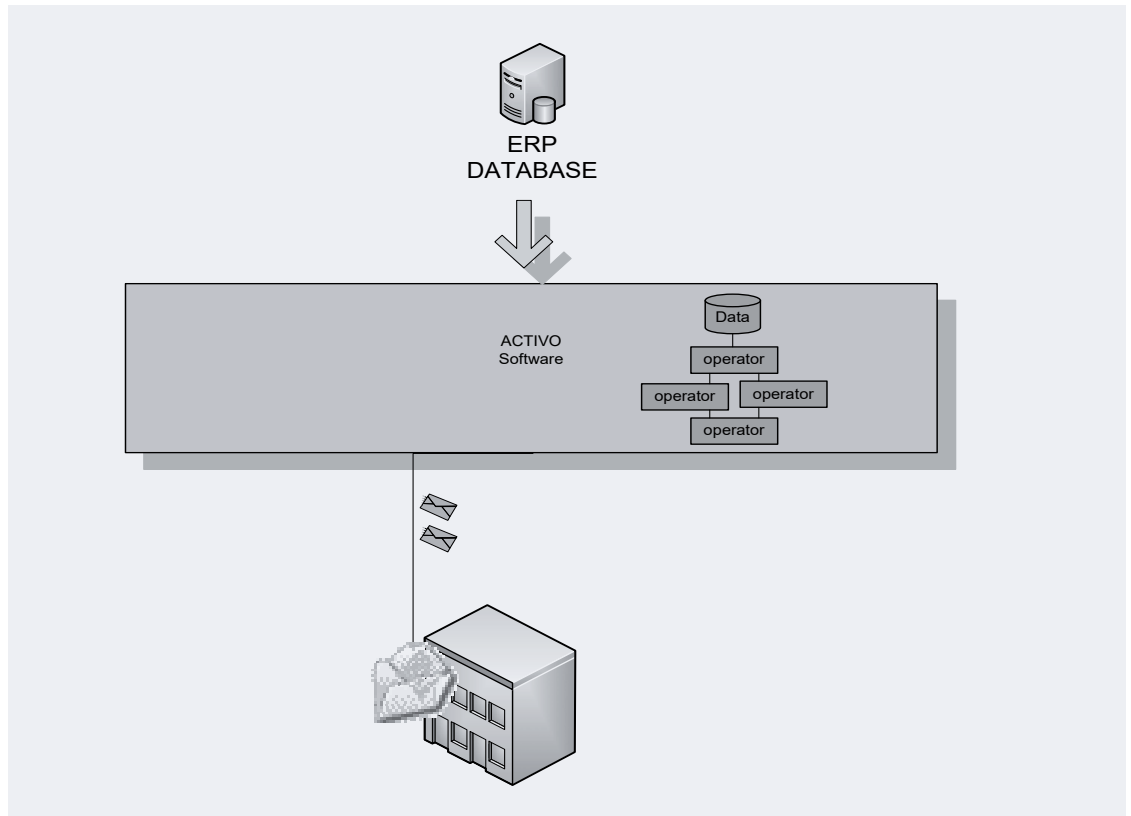
Przedsiębiorstwo chce stworzyć warstwę niezależną, pośrednią, integrującą systemy ewidencji czasu z oprogramowaniem ERP.

### PRZYKŁADOWE ZASTOSOWANIE GRAVITY

Integracja systemu ERP z dedykowanym oprogramowaniem dziedzinowym. Skonfigurowano automatyczne wywołania czasowe do przeszukiwania wskazanej lokalizacji, w poszukiwaniu plików zawierających informację o czasie pracy dla pracowników. Wywołanie operatora CALL SQL w celu umieszczenia informacji w bazie systemu kadrowo – płacowego (alternatywą jest wywołanie funkcji WEB SERVICE, jeżeli system ERP taką funkcję posiada).



## 9.8 GENEROWANIE PLIKÓW DLA URZĘDÓW PUBLICZNYCH



### KONTEKST

Przedsiębiorstwo użytkuje oprogramowanie **ERP** oraz stawia sobie za cel zautomatyzowanie komunikacji z urzędami publicznymi.

### ZADANIE

Informatyzacja komunikacji pomiędzy przedsiębiorstwem, a urzędami publicznymi stwarza konieczność szybkiej reakcji, często bez wspomaganie dostawcy systemu **ERP**.

Przykład: urząd skarbowy wprowadza konieczność dostarczania pliku informacyjnego w formacie **XML**, w jednomiesięcznym interwale czasowym.

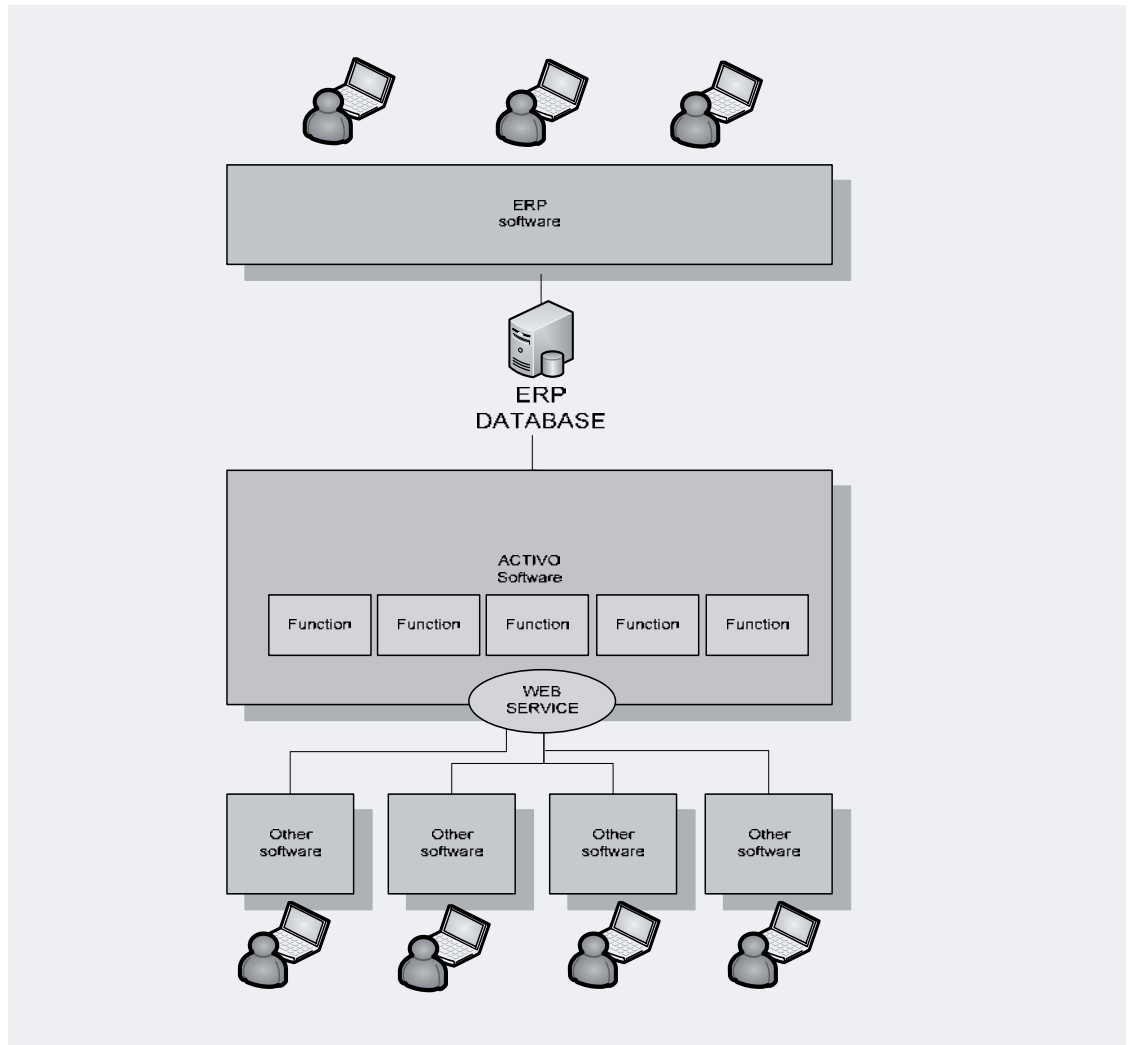
### PRZYKŁADOWE ZASTOSOWANIE GRAVITY

Proces rozpoczyna się od pobrania danych z bazy, danych systemu **ERP** (operator **INPUT SQL**) oraz wykonaniu przetworzenia. Wyjście zamodelowano jako **OUTPUT POST** (deklaracja pliku wyjściowego jako **XML**), co oznacza, że plik **XML** zostanie wysłany do wybranego adresata.

Przykładowy proces jest uruchamiany jako funkcja **WEB SERVICE**, która może być wywoływana np. z systemu **ERP**.



## 9.9 ROZBUDOWA SYSTEMU ERP



### KONTEKST

Przedsiębiorstwo używa oprogramowania ERP bez funkcji WEB SERVICE. Oprogramowanie jest „głuche”, zaś integracja może odbywać się tylko i wyłącznie na poziomie bazy SQL.

### ZADANIE

Stawiamy sobie zadanie, aby „otworzyć” oprogramowanie poprzez stworzenie funkcji WEB SERVICE, do których będą mogły odwoływać się dowolne zewnętrzne systemy informatyczne.

### PRZYKŁADOWE ZASTOSOWANIE GRAVITY

Krok pierwszy

Tworzymy funkcje, do których mogą odwoływać się systemy zewnętrzne (np. zapisz zamówienie, pobierz kartotekę indeksów, pobierz kredyty kupieckie dla kontrahentów etc.)

Krok drugi

Uruchamiamy w GRAVITY serwer **WEB SERVICE**.

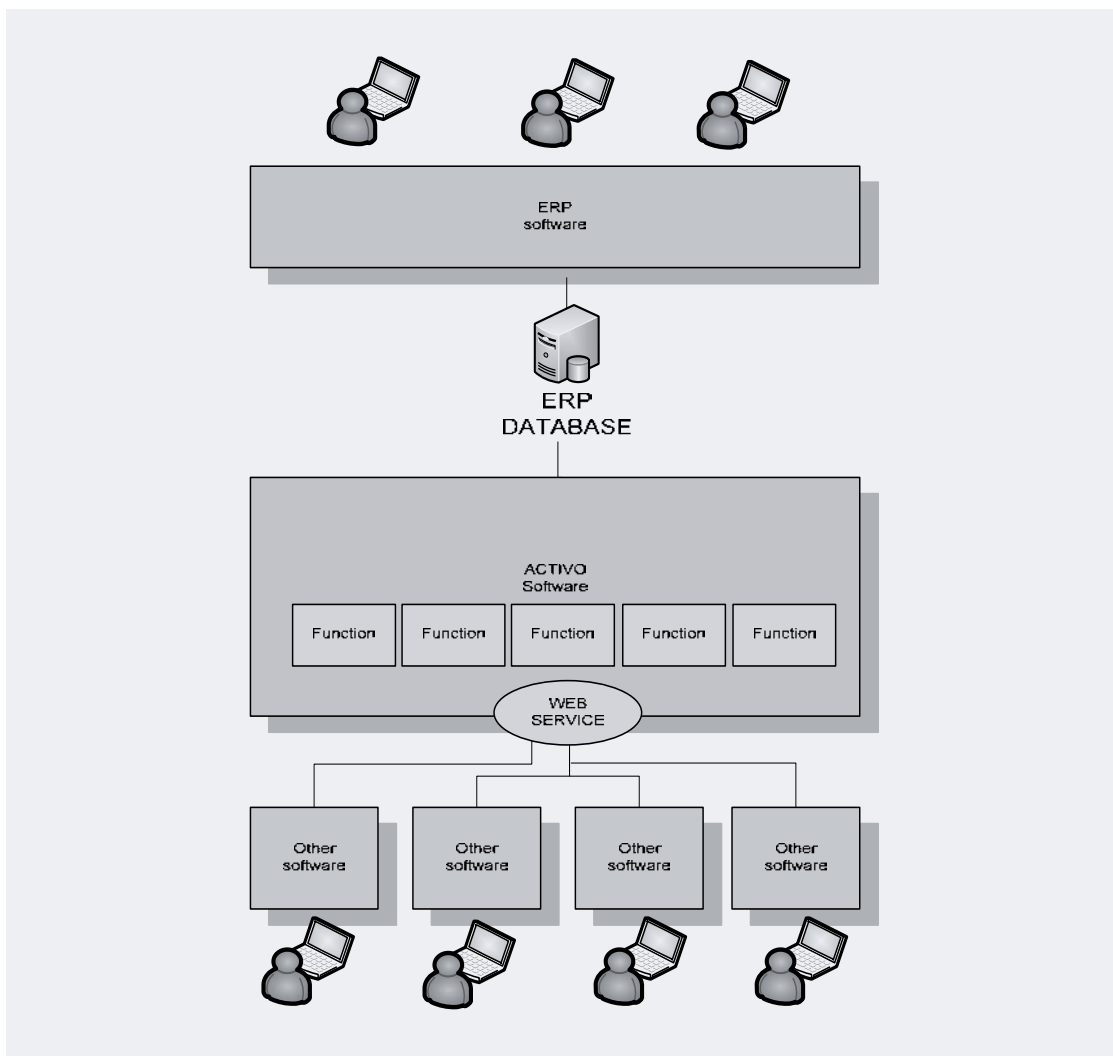
Akcja

Poprzez ten serwer wywołując funkcję **WEB SERVICE** z odpowiednim parametrem, oprogramowanie **ERP** może wywoływać przetwarzanie zdefiniowane w GRAVITY (w zakresie zdefiniowanym w kroku pierwszym).





## 9.10 RAPORTOWANIE



### KONTEKST

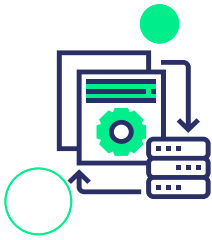
Przedsiębiorstwo użytkuje oprogramowanie, które nie posiada wielu przydatnych raportów.

### ZADANIE

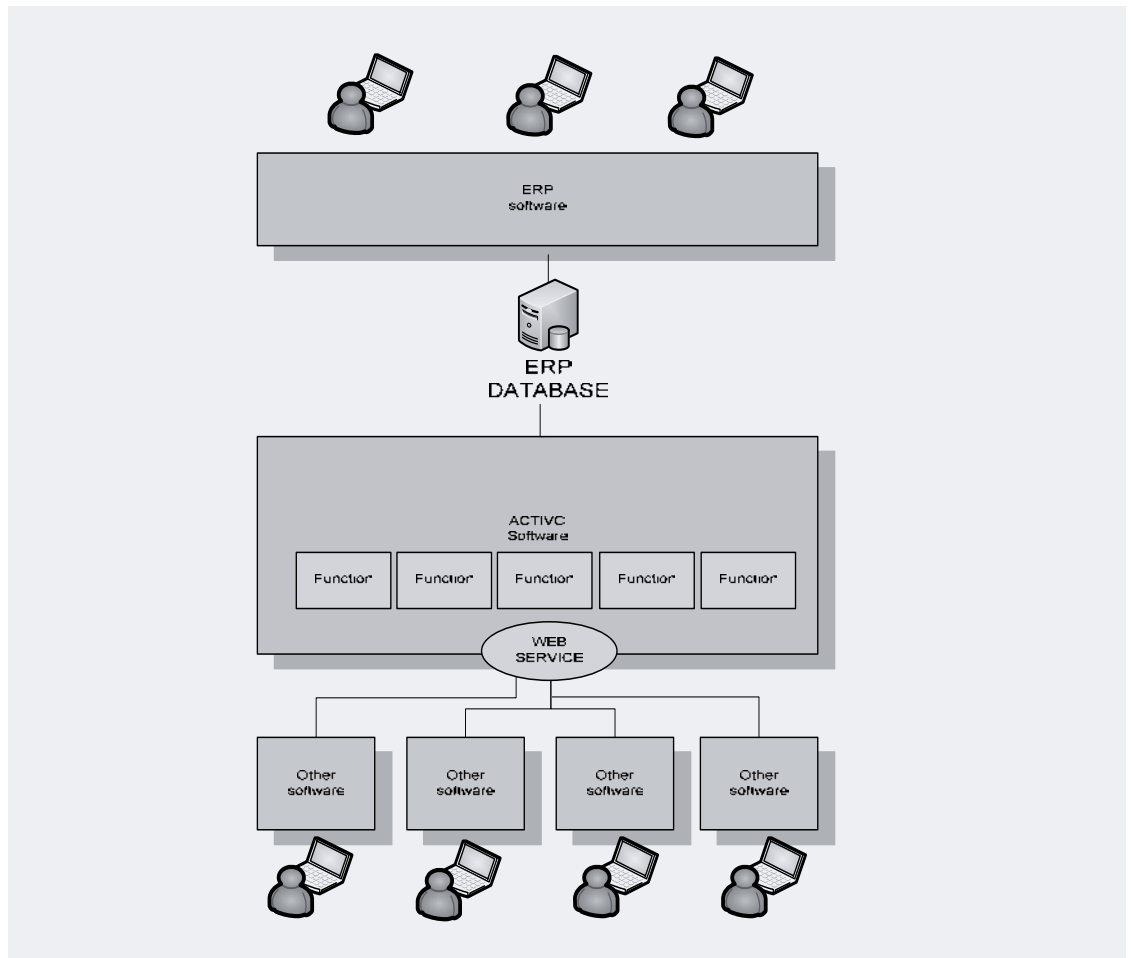
Tworzenie raportów w szacie graficznej oraz w zakresie zgodnym z zapotrzebowaniem.

### PRZYKŁADOWE ZASTOSOWANIE GRAVITY

W GRAVITY możemy stworzyć raporty, wykorzystujące strumień danych pobrane przez proces. Możemy stworzyć dowolne raporty w formacie pdf, powstające zarówno w sposób automatyczny np. w jakimś interwale czasowym, albo jako reakcja na jakieś zdarzenie, lub wywoływane poprzez oprogramowanie zewnętrzne jako funkcja **WEB SERVICE**.



## 9.11 INTEGRACJA APLIKACJI INTERNETOWYCH Z SYSTEMEM ERP



### KONTEKST

Przedsiębiorstwo użytkuje oprogramowanie **ERP** bez funkcji **WEB SERVICE**. Przedsiębiorstwo chce rozwijać rozwiązania internetowe od innego dostawcy informatyki.

### ZADANIE

Stawiamy sobie zadanie, aby 'otworzyć' oprogramowanie poprzez stworzenie funkcji **WEB SERVICE**, do których będą mogły odwoływać się dowolne zewnętrzne systemy informatyczne.

Tworząc niezależną warstwę komunikacyjną w GRAVITY, tworzymy ustrój informatyczny, w którym aplikacje frontowe mogą być dowolnie wymieniane bez czasochłonnej integracji.

Zadanie jest de facto przykładem opisanym w ogólności w punkcie powyżej (rozbudowa **ERP**).

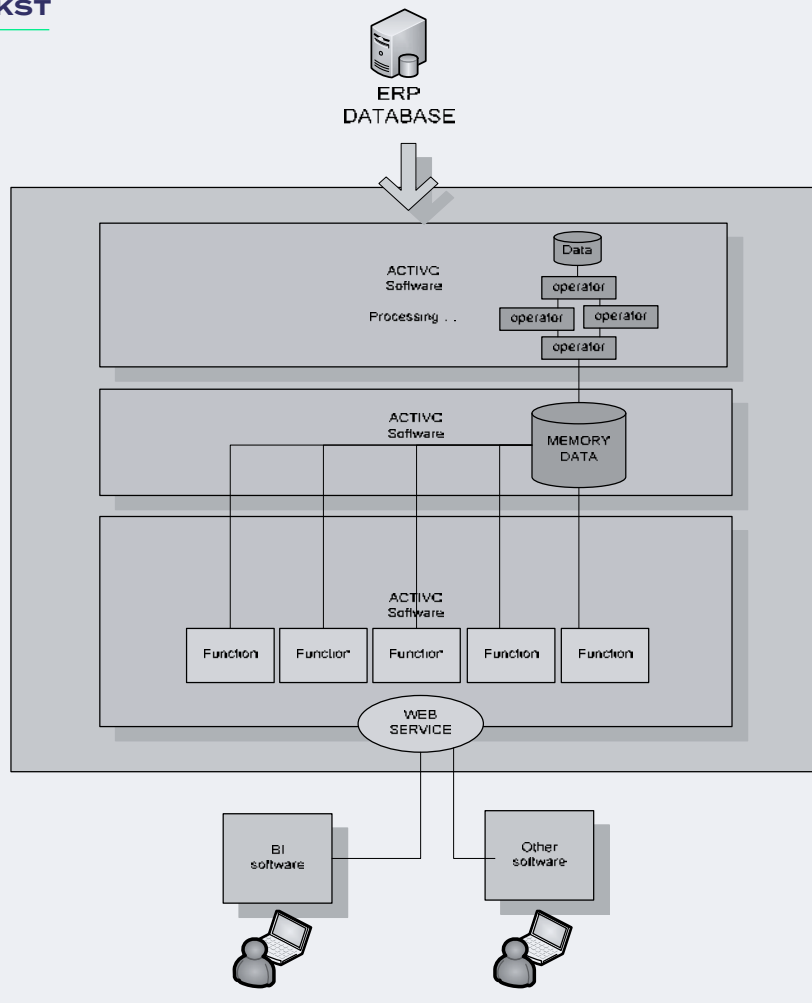
### PRZYKŁADOWE ZASTOSOWANIE GRAVITY

Zadanie jest de facto szczególnym przypadkiem opisanym w ogólności w punkcie powyżej (rozbudowa **ERP**). Wyróżniamy to zastosowanie w niezależnym przykładzie, ze względu na powszechność zapotrzebowania na taki model komunikacji.



## 9.12 PRZYSPIESZENIE PRZETWARZANIA I WYKONANIE SZYBKICH RAPORTÓW

### KONTEKST



Przedsiębiorstwo używa oprogramowania ERP, w którym zrealizowano raportowanie. Niestety istnieją przetwarzania i raporty, które ze względu na liczbę danych oraz złożoność przetwarzania wykonują się w czasie, który jest zbyt długi, co nie daje możliwości użytkownikowi na ich efektywne stosowanie w praktyce.

### ZADANIE

Przyspieszenie przetwarzania danych. Zagregowane dane dla wybranych raportów i analiz są dostępne w pamięci podręcznej.

## PRZYKŁADOWE ZASTOSOWANIE GRAVITY

### Krok 1

Stworzenie projektu, którego zadaniem jest utrzymywanie w pamięci serwera świeżo przetworzonych obliczeń i agregacji (korzyść: szybki dostęp do przetworzonych danych).

W projekcie można wykorzystać operator INPUT SQL do pobrania danych oraz operator COMPUTING do przetworzenia danych. Przetworzone, zagregowane dane są umieszczone w bazie tymczasowej, dostępnej dla innych projektów (operator OUTPUT MEMORY).

Projekt jest wywoływany przez mechanizm wywołań czasowych z ustawionym krótkim okresem interwału.

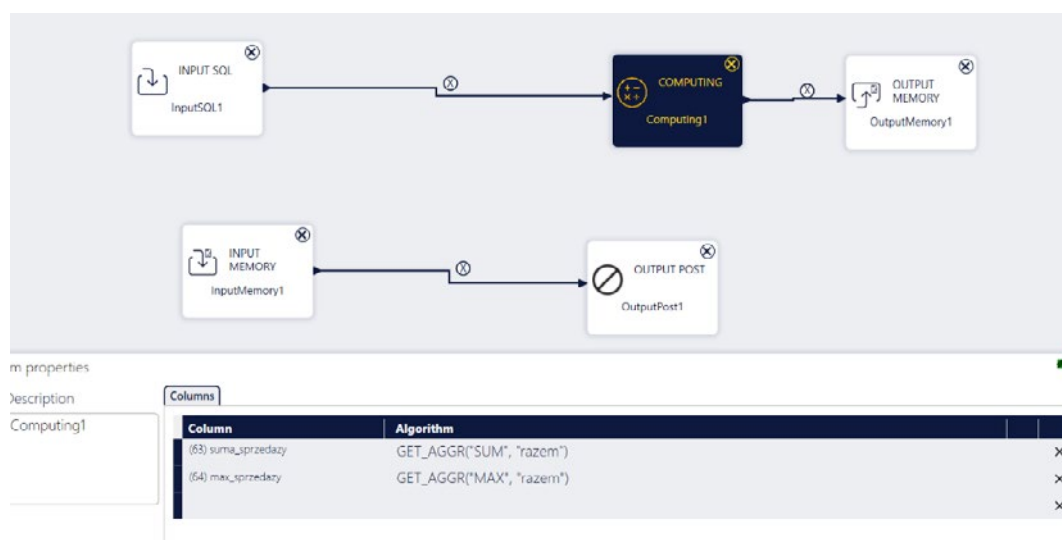
### Krok 2

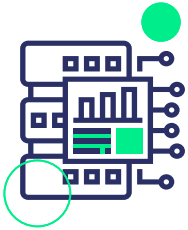
Stworzenie projektu, którego zadaniem jest wysłanie danych do prezentacji.

W projekcie wykorzystaliśmy operator INPUT MEMORY do szybkiego pobrania danych oraz operator OUTPUT DATA (wyjście danych dla funkcji WEB SERVICE)

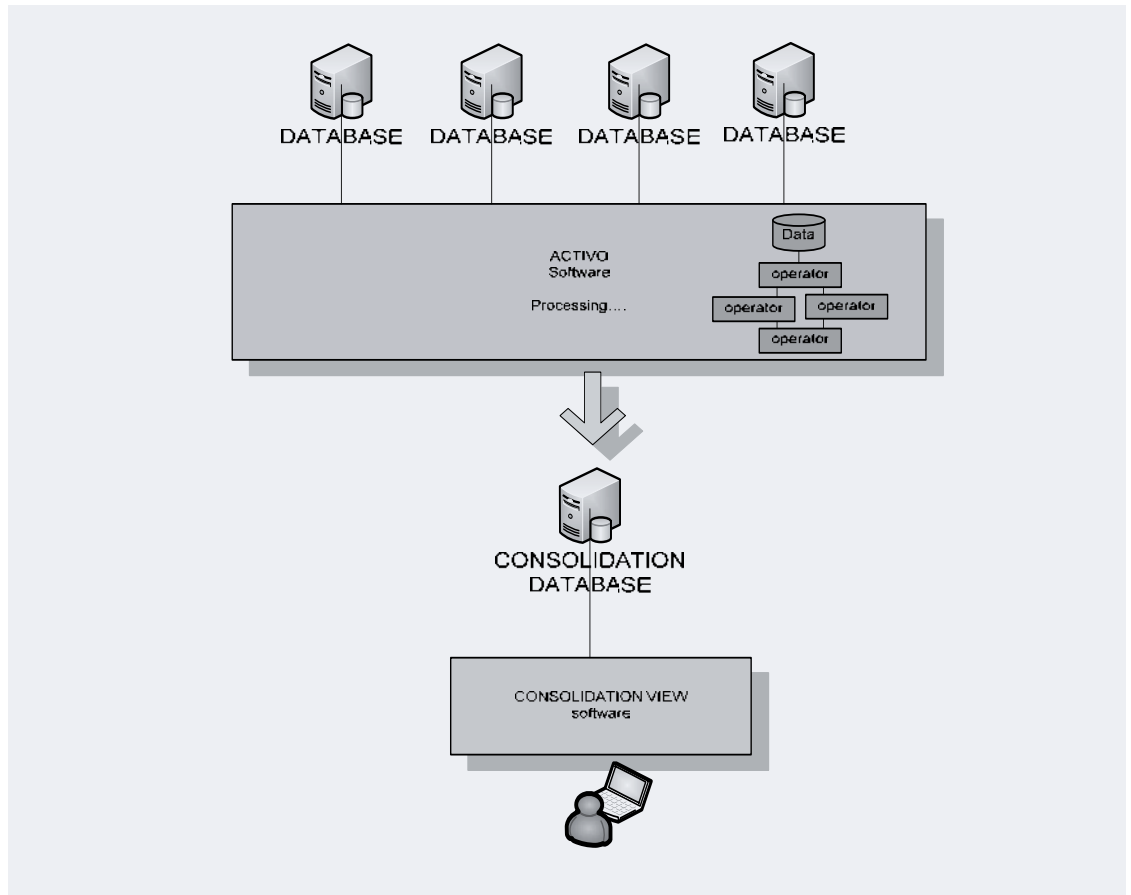
Projekt jest wywoływany jako funkcja WEB SERVICE z systemu ERP.

Diagram przykładowy





## 9.13 KONSOLIDACJA DANYCH



### KONTEKST

Przedsiębiorstwo ma wiele spółek zależnych, użytkujących różne (lub takie same) oprogramowania.

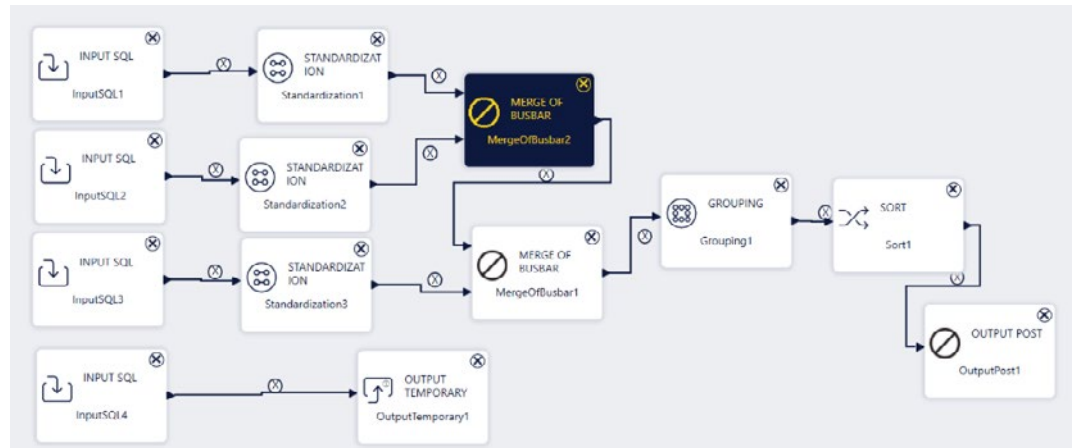
### ZADANIE

Przedsiębiorstwo chce skonsolidować dane o obrotach kontrahenckich.

### PRZYKŁADOWE ZASTOSOWANIE GRAVITY

W GRAVITY projektujemy operatory **INPUT SQL** dla wszystkich konsolidowanych baz. Za pomocą operatora **STADARIZATION**, identyfikujemy tych samych kontrahentów w różnych bazach, a obroty sumujemy. Wynik obliczeń jest transferowany do bazy wyjściowej, do której jest podłączone oprogramowanie analityczne lub wysyłane pocztą do wybranych adresatów.

Diagram przykładowy



### ZADANIE

Przedsiębiorstwo chce zgromadzić wszystkie faktury sprzedaży w jednej bazie konsolidującej, wszystkich spółek zależnych (zadanie jest rozszerzoną wersją zadania poprzedniego).

### PRZYKŁADOWE ZASTOSOWANIE GRAVITY

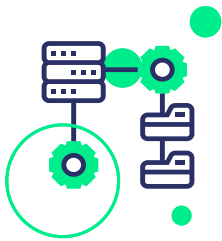
W GRAVITY projektujemy operatory **INPUT SQL** do wszystkich konsolidowanych baz. Za pomocą operatora **STANDARDIZATION**, identyfikujemy tych samych kontrahentów w różnych bazach, a obroty sumujemy. Wynik obliczeń jest transferowany do bazy wyjściowej, do której jest podłączone oprogramowanie analityczne.

### ZADANIE

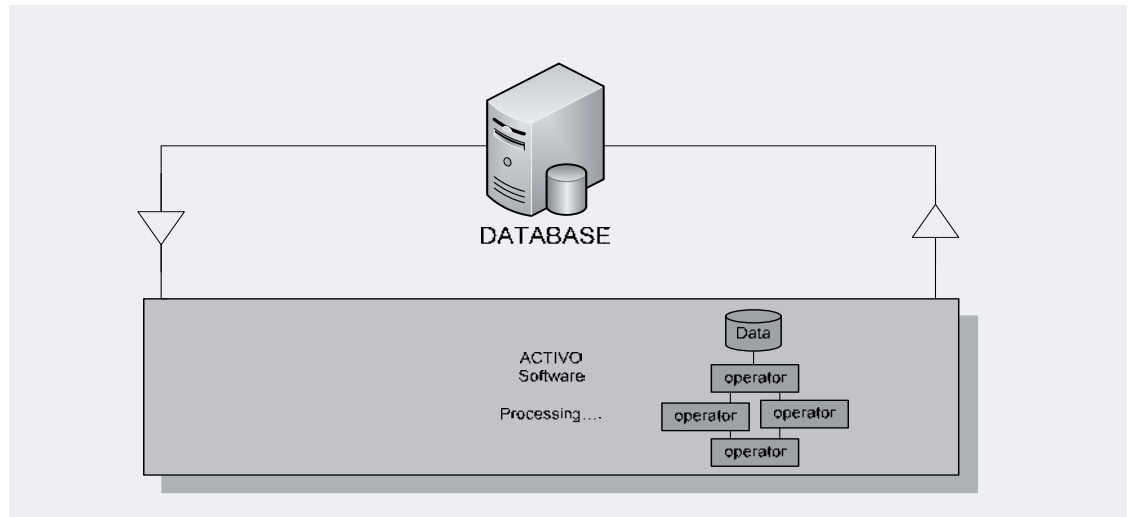
Przedsiębiorstwo chce mieć skonsolidowane arkusze ekonomiczne dla wszystkich przedsiębiorstw wchodzących w skład grupy.

### PRZYKŁADOWE ZASTOSOWANIE GRAVITY

W GRAVITY projektujemy operatory **INPUT SQL** do wszystkich konsolidowanych baz. Za pomocą operatora **STANDARDIZATION**, identyfikujemy strukturę kont księgowych w różnych bazach. Wynik obliczeń jest transferowany do bazy wyjściowej, do której podłączone jest oprogramowanie prezentujące dane (jako alternatywna opcja: wysyłanie pliku wynikowego xls na adres poczty).



## 9.14 PRZETWARZANIE DANYCH SYSTEMU ERP



### KONTEKST

Przedsiębiorstwo wykorzystuje oprogramowanie ERP z bazą SQL. W trakcie normalnej eksploatacji systemu zdarzają się sytuacje awaryjne, zarówno związane, jak i niezwiązane z użytkowaniem oprogramowaniem ERP, które wpływają na jakość, poprawność oraz integralność przechowywanych danych.

### ZADANIE

Zadaniem służb informatycznych, jest stworzenie procedur badania danych, tak, aby zminimalizować ryzyko danych niepoprawnych.

### PRZYKŁADOWE ZASTOSOWANIE GRAVITY

W GRAVITY projektujemy proces testowania rekordów w wybranych tabelach, kontrolując, czy nie zawierają niedozwolonych wartości. Gdy zostaną wykryte takie przypadki, w zależności od tabeli oraz rodzaju następuje kasowanie wadliwego rekordu lub wysłanie informacji o nieczystych danych.

### KONTEKST

Przedsiębiorstwo wykorzystuje oprogramowanie ERP z bazą SQL. W trakcie użytkowania systemu ERP, zaobserwowano pojawianie się nieintencjonalnych duplikatów np. w słowniku kontrahentów.

### ZADANIE

Testowanie bazy i likwidacja potencjalnych duplikatów.

### PRZYKŁADOWE ZASTOSOWANIE GRAVITY

W GRAVITY projektujemy proces testowania rekordów w tabeli z danymi kontrahentów, badając, czy nie istnieją duplikaty, czyli rekordy zawierające pola podobne z pewnym prawdopodobieństwem (operator TEXT RECON). W przypadku wykrycia takich przypadków, umieszczamy w tabeli odnośnik do rekordu macierzystego (oczywiście, jeżeli oprogramowanie ERP na to pozwala). Jeżeli oprogramowanie ERP na to nie pozwala, informację o wynikach rozpoznania możemy umieścić w pliku i wysłać, lub umieścić w zadeklarowanej bazie (dostęp przez oprogramowanie zewnętrzne).

### **KONTEKST**

Przedsiębiorstwo wykorzystuje oprogramowanie ERP. Użytkownik chciałby poszerzyć badanie integralności danych poprzez badanie reguł, który mógłby w dowolny sposób definiować.

### **ZADANIE**

Kontrola reguł księgowych, w zakresie zaksięgowania obrotów na poszczególnych kontach

### **PRZYKŁADOWE ZASTOSOWANIE GRAVITY**

W GRAVITY projektujemy proces badanie reguły mówiącej, że suma obrotów kont księgowych zespołu 5 musi się pokrywać z sumą kontrolną, zaksięgowaną w obrotach konta 490.

### **KONTEKST**

Przedsiębiorstwo wykorzystuje oprogramowanie ERP oraz oprogramowanie analityczne np. klasy BI.

### **ZADANIE**

Przygotowanie danych systemu ERP do szybkiego zastosowania, dla oprogramowania analitycznego klasy BI.

### **PRZYKŁADOWE ZASTOSOWANIE GRAVITY**

W GRAVITY projektujemy proces pobierania danych oraz obliczeń sum agregacyjnych. Sumy agregacyjne oraz potencjalne duplikacje są utrzymywane w pamięci podręcznej, zapewniając oprogramowaniu BI dostęp do przygotowanych danych.

Proces jest wywoływany przez mechanizm wywołań czasowych z krótkim interwałem, dzięki temu w pamięci podręcznej rezydują zawsze najnowsze obliczenia.

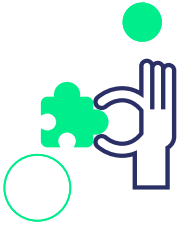
### **PRZYKŁADOWE ZASTOSOWANIE GRAVITY**

W GRAVITY projektujemy proces pobierania danych oraz normalizacji jednostek, na użytek sumowania lub konsolidacji danych..

### **PRZYKŁADOWE ZASTOSOWANIE GRAVITY**

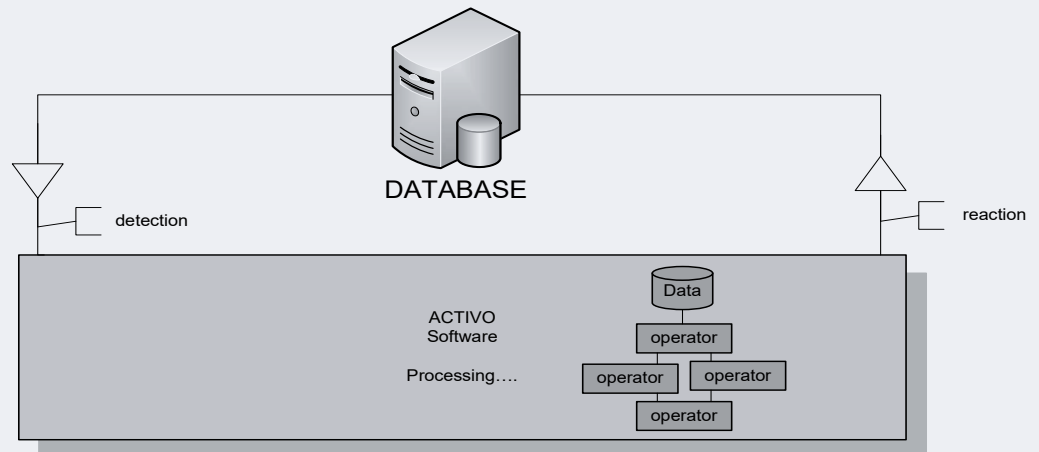
W GRAVITY projektujemy proces pobierania danych oraz spajania różnych baz danych.





## 9.15 ŚLEDZENIE ZMIAN DANYCH KRYTYCZNYCH

### KONTEKST



Przedsiębiorstwo używa oprogramowania ERP, w którym nie ma możliwości śledzenia zmian dokonywanych w jakimś krytycznym dla przedsiębiorstwa rodzaju zdarzenia.

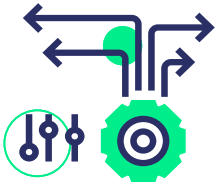
### ZADANIE

Przedsiębiorstwo chce wprowadzić system śledzenia wszystkich modyfikacji dla wybranego zdarzenia (np. edycji dokumentu rozchodu wewnętrznego).

### PRZYKŁADOWE ZASTOSOWANIE GRAVITY

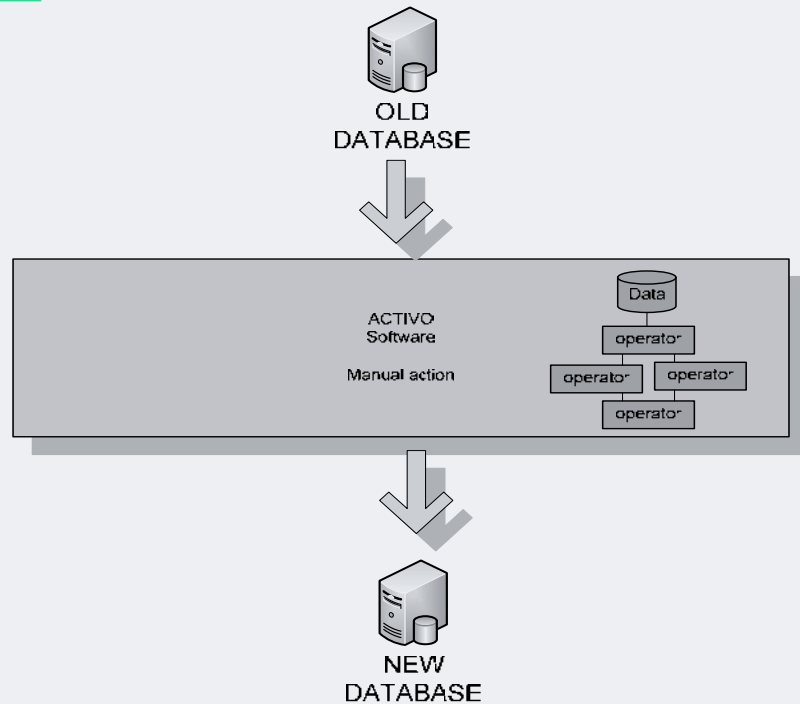
Projekt jest wywołany przez notyfikację bazy danych, w związku z edycją wskazanej tabeli (zawierającej dane rozpatrywanego zdarzenia).

Projekt po wywołaniu, zapamiętuje w innej tabeli dane przed edycją oraz dane po edycji wraz z czasem modyfikacji.



## 9.16 KONWERSJA DANYCH ZEWNĘTRZNYCH – MIGRACJA DANYCH

### KONTEKST



Przedsiębiorstwo jest w trakcie wdrożenia nowego oprogramowania. W bazie danych starego oprogramowania są dane, które należy wytransferować do nowego systemu.

### ZADANIE

Przedsiębiorstwo musi stworzyć projekt migracji danych technologii produkcyjnej, który może być wielokrotnie używany do celów pilotażowych oraz startu produkcyjnego.

### PRZYKŁADOWE ZASTOSOWANIE GRAVITY

Model projektu zawiera akcje pobrania kartoteki materiałów, kartoteki produktów, marszruty technologicznej oraz receptury materiałowej.

Zakładamy, że dane wejściowe są zgromadzone w plikach xls, zaś baza wyjściowa hipotetycznego oprogramowania ERP jest bazą SQL.

Projekt jest wywoływany manualnie przez użytkownika.

### ZADANIE

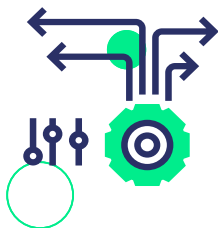
Przedsiębiorstwo musi stworzyć projekt migracji danych kadrowo – płacowych, który może być wielokrotnie używany do celów pilotażowych oraz startu produkcyjnego.

### PRZYKŁADOWE ZASTOSOWANIE GRAVITY

Model projektu zawiera akcje pobrania kartoteki pracowników, umów o pracę oraz listy płac. Zakładamy, że dane wejściowe są zgromadzone w bazie **SQL** dotychczasowego systemu, zaś baza wyjściowa hipotetycznego oprogramowania **ERP** jest bazą **SQL**.

Projekt jest wywoływany manualnie przez użytkownika.

## 9.17 ZASTOSOWANIE OPERATORA OUTPUT SQL



Operator **OUTPUT SQL** jest komponentem najbardziej złożonym i z tego też powodu wymagającym szczególnej uwagi. Poniżej przygotowaliśmy dla Ciebie kilka przykładów, które pozwolą lepiej zrozumieć jego zastosowanie.

### KONTEKST

Przedsiębiorstwo posiada wiele spółek zależnych, użytkujących różne oprogramowania lub to samo oprogramowanie, jednak składających dane w różnych bazach.

### ZADANIE

Przedsiębiorstwo chce gromadzić wszystkie dane o sprzedaży w jednej bazie, celem konsolidacji oraz analiz typu **BI**.

Sformułowane przez nas zadanie użycia GRAVITY wpisuje się w przykłady opisane w niniejszym rozdziale, w części oznaczonej jako **KONSOLIDACJE**.

W poniższych przykładach skupimy się tylko i wyłącznie na sposobach zastosowania operatora **OUTPUT SQL**.

Przykładowe zastosowanie operatora **OUTPUT SQL**

Interakcja z danymi źródłowymi

W niniejszym przykładzie przedstawiamy konfigurację z założeniem interakcji z bazami źródłowymi.

Standaryzacja

Ponieważ konsolidujemy dane z różnych źródeł, unikalne identyfikatory muszą być podczas akcji zapisu standaryzowane (zmieniane tak, aby uniknąć duplikatów).

Strumień danych wejściowych

Podczas zapisu każdego rekordu, mamy możliwość ustawienia flagi poboru danych, w każdej bazie źródłowej. Dzięki temu operatory **INPUT SQL** (tworzenie strumienia danych wejścia), możemy budować z wykorzystaniem warunku ograniczającego pobór strumienia danych tylko do tych, które nie były wcześniej pobierane.

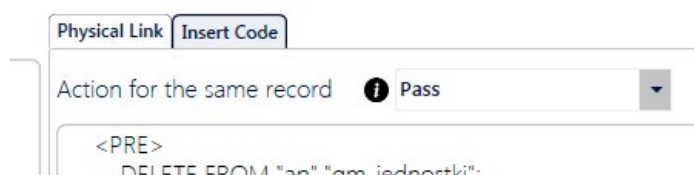
Na ilustracji poniżej wskazaliśmy, że w takim wypadku nastawa **ACTION FOR THE SAME RECORD** musi być ustawiona na **'PASS'**.

Na górnej belce tabeli, wskazanej jako tabela 'celu', musisz wskazać akcję **STANDARIZATION** (patrz ilustracja poniżej). Po wyborze akcji **STANDARIZATION** system poprosi o informację o identyfikatorze źródła (zważ, że w słowniku standaryzacyjnym może być wiele źródeł danych: musisz zatem nadać nazwę swojemu źródłu lub wskazać bit magistrali wejściowej, która to zawiera unikalną nazwę).

Aby ustawiać flagę na danych źródłowych, musisz wskazać w obszarze **SQL ACTIONS** akcję **SETUP INPUT** (patrz ilustracja poniżej) dla każdego.

Aby zakończyć konfigurację akcji **INPUT SQL** musisz wskazać (patrz ilustracja poniżej):

**INPUT**.....wskaz operator typu **INPUT** skąd pobierasz dane do magistrali



**INPUT TABLE**.....wskaż tabelę źródłową z operatora **INPUT**, którą chcesz edytować

**COLUMN FOR UPDATE**.....wskaż kolumnę z tabeli źródłowej, która będzie edytowana (miejsce ustawienia flagi)

**VALUE FOR UPDATE**.....wskaż wartość edytowaną (wartość flagi)

**ID SOURCE RECORD**.....wskaż sposób identyfikacji rekordu; wskazujesz kolumnę unikalną oraz kolumnę z magistrali wejściowej do operatora **OUTPUT SQL**;

#### Uwaga!



Możesz skonfigurować przepływ strumienia danych dla każdego źródła niezależnie, co oznacza, że będziesz miał tyle operatorów **INPUT SQL** oraz operatorów **OUTPUT SQL** ile źródeł.

Możesz również zadanie zrealizować inaczej: w takim wypadku proces będzie zasilany z tylu **INPUT SQL**, ile jest źródeł danych, lecz dane są łączone do jednej magistrali (używając np. operatora **MERGE OF BUSBAR**), a następnie jest konfigurowany tylko jeden operator **OUTPUT SQL**. W takim jednak przypadku musisz w operatorze **OUTPUT SQL** wskazać tyle akcji (**ACTIONS SQL**) typu **SETUP INPUT**, ile jest źródeł (dla każdego źródła niezależnie jedna akcja) wraz z dodaniem warunku wykonania akcji (istnieje możliwość wskazania warunku realizacji akcji **SETUP INPUT** w zależności od wartości bitu (kolumny) strumienia danych).

#### Uwaga!



Powyższy przykład jest skuteczny dla obiektów (w naszym zadaniu obiektem są faktury), które podlegają zmianom, ale nie podlegają kasowaniu w źródle (faktury są co najwyżej korygowane).

## PRZYKŁADOWE ZASTOSOWANIE OPERATORA OUTPUT SQL

Interakcja z danymi źródłowymi

W niniejszym przykładzie zakładamy brak możliwości interakcji z bazami źródłowymi.

Standaryzacja

Ponieważ konsolidujemy dane z różnych źródeł, unikalne identyfikatory muszą być podczas akcji zapisu standaryzowane (zmieniane tak, aby uniknąć duplikatów).

Ponieważ zastosowanie operatora **OUTPUT SQL** wiąże się z operatorami **INPUT SQL**, spróbujmy rozważyć różne sposoby współpracy tych operatorów. Poniżej przestudiowaliśmy skutek, w zależności od nastaw konfiguracyjnych operatora źródłowego **INPUT SQL**, w obszarze zakresu pobierania danych i nastaw konfiguracyjnych operatora **OUTPUT SQL** w zakresie akcji **SQL**, dotyczących kasowania starych danych w tabeli celu.

INPUT SQL OUTPUT SQL	Pobieranie całego zakresu danych	Pobieranie inkrementacyjne	Pobieranie inkrementacyjne z 'tough position'
<p>Kasowanie wszystkich rekordów w tabeli celu</p> <p>Akcja SQL</p> <p>* DELETE ALL RECORDS</p> <p>* DELETE ALL RECORDS FOR SOURCE</p>	<p>Zalety:</p> <ul style="list-style-type: none"> <li>– zawsze pełna i jednoznaczna informacja w tabeli celu</li> </ul> <p>Wady:</p> <ul style="list-style-type: none"> <li>– czasochłonność akcji</li> </ul>	<p>Zabronione.</p> <p>Nie możesz pobierać inkrementacyjnie i kasować całej informacji w tabeli celu, ponieważ nie będziesz miał pełnej informacji.</p>	<p>Zabronione.</p> <p>Patrz uwaga obok.</p>
<p>Kasowanie rekordów w tabeli celu w zakresie deklarowanym na stałe</p> <p>Akcja SQL</p> <p>* DELETE DECLARED SPACE</p>	<p>Identycznie jak wyżej.</p> <p>Akcja różni się od przykładu powyżej tym, że zakładamy przestrzenie identyfikacyjne dla każdego niezależnego źródła, co powoduje, że nie jest potrzebna standaryzacja.</p> <p>Przypadek bez standaryzacji!</p>	<p>Zabronione.</p> <p>Patrz uwaga powyżej.</p>	<p>Zabronione.</p> <p>Patrz uwaga powyżej.</p>
<p>Kasowanie rekordów w tabeli celu w zakresie wynikającym ze strumienia danych wejściowych</p> <p>Akcja SQL</p> <p>* DELETE SPACE BY MIN MAX</p>	<p>Możliwe, jednak ryzykowne.</p> <p>Uzasadnienie: mogą istnieć rekordy spoza zakresu wynikającego ze strumienia danych, które w źródle pierwotnym zostały skasowane (w tym wypadku w tabeli celu nie zostaną usunięte).</p>	<p>Warunek:</p> <ul style="list-style-type: none"> <li>– rekordy strumienia wejściowego już pobrane nie mogą być kasowane lub zmieniane</li> </ul> <p>Zalety:</p> <ul style="list-style-type: none"> <li>– szybkość przetwarzania</li> </ul>	<p>Warunek:</p> <ul style="list-style-type: none"> <li>– rekordy strumienia wejściowego z identyfikatorami poniżej 'twardej pozycji' nie mogą być kasowane lub zmieniane</li> </ul> <p>Zalety:</p> <ul style="list-style-type: none"> <li>– szybkość przetwarzania</li> </ul>
<p>Kasowanie lub zmiana rekordu powtórnie pobranego</p> <p>Akcja SQL</p> <p>* DELETE OLD RECORD</p> <p>* UPDATE OLD RECORD</p>	<p>Identycznie jak wyżej</p>	<p>Skutek akcji identyczny do akcji powyżej</p>	<p>Nie stosować.</p> <p>Skutek akcji identyczny jak powyżej pogorszony o sytuację, w której mamy rekordy skasowane w strumieniu wejściowym, w zakresie powyżej twardej pozycji, jednak wcześniej pobrane.</p>



CREATED BY

**CAI**  
**CAFFEINE**  
**MINDS**

FUTURE TECH FORMULA

Tworzymy oprogramowanie  
ERP oraz BI od 20 lat.