Kamień milowy numer 2

Szymon Gut
Jan Krężel

May 2023

1 Korzyści z perspektywy odbiorcy oraz cel projektu

Rozwiązanie, które pragnęlibyśmy przedstawić obejmuje stworzenie hurtowni danych oraz dołączenie któregoś z narzędzi Buissness Inteligence dla danych statystycznych pochodządzych z Głównego Urzędu Statystycznego dotyczących rozwoju gmin w Polsce na przestrzeni lat. Projekt ma na celu wskazanie oraz wyróżnienie gmin prężnie rozwijających się jako potencjalne do planowanych inwestycji. Umożliwi to kompleksową analizę rozwoju gmin na przestrzeni lat, co pozwoli na określenie tendencji i prognozowanie przyszłych potrzeb. Zbudowanie hurtowni danych pozwoli na łatwiejszą identyfikację problemów i potrzeb mieszkańców, co z kolei umożliwi urzędnikom gminnym skuteczniejsze planowanie działań.

Poprzez porównywania danych pomiędzy różnymi gminami, możliwe jest wymiana dobrych praktyk oraz efektywne planowanie wspólnych działań. Dodatkowo zestawienie metryk wskazujących na rozrost gmin pozwoli na weryfikację władz samorządowych, czy te wywiązują się ze swoich obietnic.

Dołączenie narzędzi Business Intelligence umożliwi dokładniejszą analizę zebranych danych oraz pozwoli na kompleksowe raportowania wyników działań władz gminnych wobec władz centralnych oraz badanie ich skuteczności. Wspomniana analiza pozwoli na wczesne wykrywanie nieprawidłowości, np. w obszarze polityki społecznej czy planowania przestrzennego.

2 Diagram Proponowanej Architektury Rozwiązania

Na Rysunku 1 widoczny jest uproszczony diagram proponowanej architektury. Dane, z których korzystać będzie nasza hurtowania będą pobierane za pomocą API z Banku Danych Lokalnych Głównego Urzędu Statystycznego.

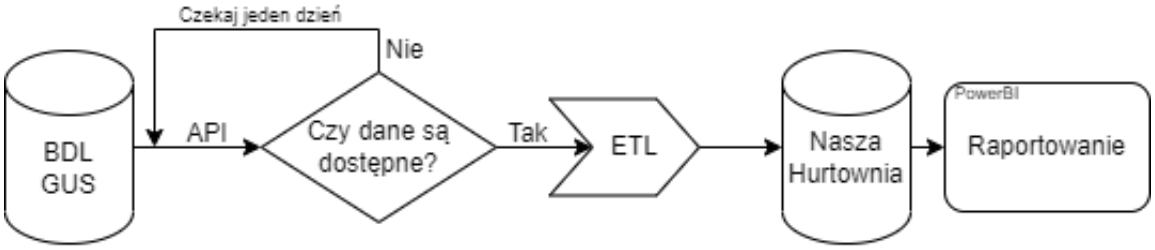
To API podlega pewnym limitom zapytań jednak są one dość niewielkie w stosunku do częstości pobierania danych – dane w naszej hurtowni wymagają odświeżania raz na rok i pochodzą z relatywnie niewielu źródeł, a najbardziej restrykcyjny limit zapytań wynosi 10000 zapytań w ciągu tygodnia.

Dane po pobraniu będą przekazywane do procesu ETL, który dokładniej opisany jest w późniejszej sekcji. Po przekształceniu i załadowaniu dane są gotowe do użycia z poziomu naszej hurtowni.

Docelowo cały proces powinien być zautomatyzowany. To znaczy dane powinny być pobierane, przekształcane i ładowane do hurtowni całkowicie bez potrzeby ingerencji człowieka na żadnym z tych etapów. Ponadto, nasz zautomatyzowany proces nie powinien "bezmyślnie" raz do roku próbować ściągać dane i ponownie wchodzić w stan uśpienia. Proces powinien wpierw sprawdzać czy najnowsze dane są już dostępne. W przypadku gdy dane nie są jeszcze dostępne powinno mieć miejsce ponowne sprawdzenie dostępności (na przykład) kolejnego dnia do momentu, aż dane będą gotowe do pobrania.

3 Pozyskane dane

Dane zostały pozyskane z oficjalnej strony Głównego Urzędu Statystycznego. Wszystkie wykorzystane zbiory danych dostępne są powszechnie oraz niedopłatnie. Częstotliwość odświeżania użytych danych jest równa jednemu rokowi kalendarzowemu.



Rysunek 1: Uproszczony Diagram Architektury

Do analiz wykorzystano dane dotyczące średniego dochodu na 1 mieszkańca dla danych gmin, wydatki gminy w przeliczeniu na 1 mieszkańca, liczbę nowych budynków mieszkalnych oddanych do użytku w danych gminach, rejestrowane bezrobocie oraz liczbę bibliotek w danych gminach i na sam koniec inflację w Polsce w danym roku. Łącznie wykorzystano zatem 6 różnych zbiorów danych, które w dalszej fazie przekształcono procesem ETL i załadowano do hurtowni.

3.1 Dochody na 1 mieszkańca

Przeanalizowano dochody gminy przeliczone na jednego mieszkańca. Ukazuje to rozwój powierzchni przemysłowych oraz zamożność mieszkańców gminy, gdyż jednym z dochodów są płacone w tej gminie podatki.

W tym celu wykorzystano zbiór danych pochodzący z **Finanse Publiczne/Dochody budżetów gmin i miast na prawach powiatu/Dochody na 1 mieszkańca**. Zakres dostępnych lat to 2021-2002. Jako wskaźnik zostało wybrane dochody na 1 mieszkańca ogółem to jest sumaryczna wartość dochody generowana przez daną gminę.

W skład wspomnianych danych wchodzą następujące atrybuty:

Nazwa pola	**Opis**
Kod	Identyfikator jednostki terytorialnej. Pierwsze dwa znaki kodują symbol województwa (00, 02, ..., 32), dwa kolejne znaki kodują symbol powiatu (00, 01, ...99) oraz trzy ostatnie znaki kodują symbol gminy (000, ...).
Nazwa	Pełna polska nazwa danej gminy.
Rok	Rok w którym rejestrowano dane zmiany.
Wartość	Wartość obserwowanej zmiany.
Jednostka	Jednostka w której weryfikowano dane zmiany (tutaj zł).

3.2 Wydatki na 1 mieszkańca

Przeanalizowano wydatki jakie gmina ponosi na jednego mieszkańca. Odzwierciedla to rozwój oraz inwestycje gminy w celu lepszego zagospodarowania wolnych powierzchni oraz rozwój poszczególnych regionów.

W tym celu wykorzystano zbiór danych pochodzący z **Finanse Publiczne/Wydatki budżetów gmin i miast na prawach powiatu/Wydatki na 1 mieszkańca**. Zakres dostępnych lat to 2021-2001. Jako wskaźnik zostało wybrane wydatki na 1 mieszkańca ogółem to jest sumaryczna wartość wydatków poniesionych przez daną gminę na jednego mieszkańca.

W skład wspomnianych danych wchodzą następujące atrybuty:

Nazwa pola	Opis
Kod	Identyfikator jednostki terytorialnej. Pierwsze dwa znaki kodują symbol województwa (00, 02, ..., 32), dwa kolejne znaki kodują symbol powiatu (00, 01, ...99) oraz trzy ostatnie znaki kodują symbol gminy (000, ...).
Nazwa	Pełna polska nazwa danej gminy.
Rok	Rok w którym rejestrowano dane zmiany.
Wartość	Wartość obserwowanej zmiany.
Jednostka	Jednostka w której weryfikowano dane zmiany (tutaj zł).

3.3 Wkaźnik bezrobocia w danej gminie

Przeanalizowano wskaźnik bezrobocia w danych gminach. W ten sposób można uzyskać informacje na temat mieszkańców oraz stanu rozwoju gminy, gdyż wraz z rozwojem powinno rosnąć zapotrzebowanie na pracowników oraz ilość dostępnych ofert pracy.

W tym celu wykorzystano zbiór danych pochodzący z **Rynek pracy/Bezrobocie rejestrowane/Bezrobotni zarejestrowani wg płci w gminach**. Zakres dostępnych lat to 2022-2003.

W skład wspomnianych danych wchodzą następujące atrybuty:

Nazwa pola	Opis
Kod	Identyfikator jednostki terytorialnej. Pierwsze dwa znaki kodują symbol województwa (00, 02, ..., 32), dwa kolejne znaki kodują symbol powiatu (00, 01, ...99) oraz trzy ostatnie znaki kodują symbol gminy (000, ...).
Nazwa	Pełna polska nazwa danej gminy.
Płeć	Zmiana z agregacją dla danej płci (dostępne opcje: mężczyzna, kobieta)
Rok	Rok w którym rejestrowano dane zmiany.
Wartość	Wartość obserwowanej zmiany.
Jednostka	Jednostka w której weryfikowano dane zmiany (tutaj osoba).

3.4 Liczba bibliotek w danej gminie

Wykorzystano liczność bibliotek publicznych w danych gminach. Jest to nierozłączna charakterystyka świadcząca o rozwoju kulturowym gminy. Używając tego zbioru danych mamy zamiar sprawdzić, czy gmina inwestuje w zagadnienia kultorowe oraz dba o komfort swoich mieszkańców.

W tym celu wykorzystano zbiór danych pochodzący z **Kultura/Biblioteki/Biblioteki publiczne**. Zakres dostępnych lat to 2021-1995.

W skład wspomnianych danych wchodzą następujące atrybuty:

Nazwa pola	Opis
Kod	Identyfikator jednostki terytorialnej. Pierwsze dwa znaki kodują symbol województwa (00, 02, ..., 32), dwa kolejne znaki kodują symbol powiatu (00, 01, ...99) oraz trzy ostatnie znaki kodują symbol gminy (000, ...).
Nazwa	Pełna polska nazwa danej gminy.
Rok	Rok w którym rejestrowano dane zmiany.
Wartość	Wartość obserwowanej zmiany.
Jednostka	Jednostka w której weryfikowano dane zmiany (tutaj obiekt).

3.5 Liczba oddanych mieszkań do użytkowania

Wykorzystano zbiór danych zawierający liczność oddanych mieszkań do użytkowania. Wzrost liczby oddanych mieszkań może świadczyć o prężnym rozwoju gminy, z przyczyny napływu dużej ilości nowych mieszkańców oraz rozwój działalności deweloperskich.

W tym celu wykorzystano zbiór danych pochodzący z **Przemysł i budownictwo/Budownictwo mieszkaniowe/Mieszkania oddane do użytkowania**. Zakres dostępnych lat to 2021-1995.

W skład wspomnianych danych wchodzą następujące atrybuty:

Nazwa pola	Opis
Kod	Identyfikator jednostki terytorialnej. Pierwsze dwa znaki kodują symbol województwa (00, 02, ..., 32), dwa kolejne znaki kodują symbol powiatu (00, 01, ...99) oraz trzy ostatnie znaki kodują symbol gminy (000, ...).
Nazwa	Pełna polska nazwa danej gminy.
Rok	Rok w którym rejestrowano dane zmiany.
Wartość	Wartość obserwowanej zmiany.

3.6 Wskaźnik inflacji w Polsce

Wykorzystano również zbiór danych zawierający wartość inflacji w Polsce w poszczególnych latach, który wykorzystano przy budowaniu tabeli Rok jako tabeli wymiaru. Współczynnik inflacji może mieć wpływ na decyzje inwestycyjne oraz na aktualną strategię gospodarczą. Zbiór danych zawiera dokładne wskaźniki cen towarów i usług konsumpcyjnych w województwach w Polsce na przestrzeni lat. W celu wyliczenia średniej inflacji dla kraju do tabeli roku będącej wymiarem dodajemy średnią wartość tego wskaźnika w danym roku.

W tym celu wykorzystano zbiór danych pochodzący z **Ceny/Wskaźnik cen/Wskaźnik cen towarów i usług konsumpcyjnych**. Zakres dostępnych lat to 2022-2003.

W skład wspomnianych danych wchodzą następujące atrybuty:

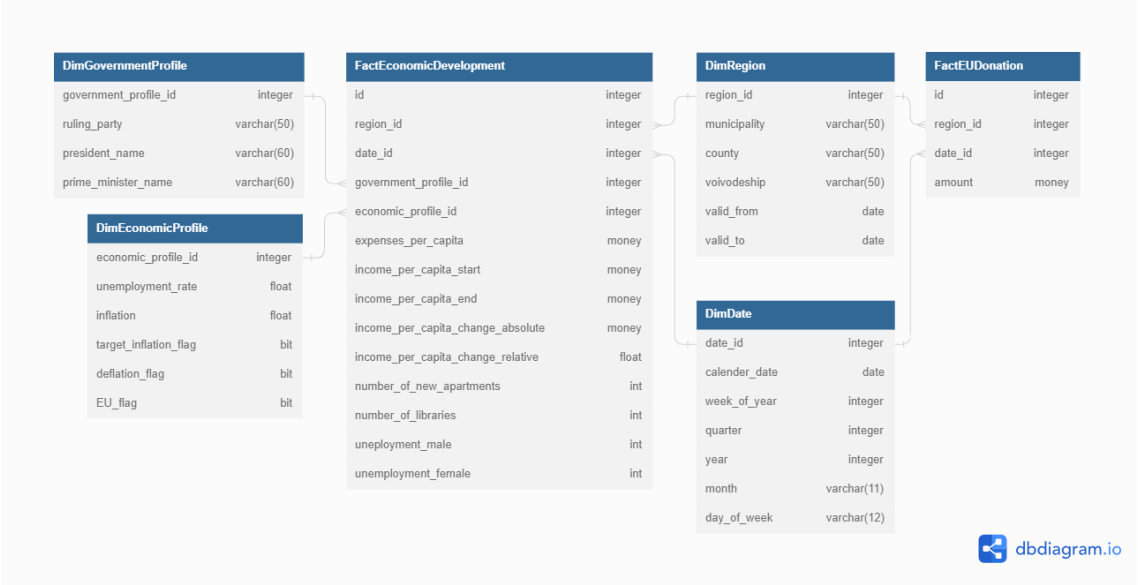
Nazwa pola	Opis
Kod	Identyfikator jednostki terytorialnej. Pierwsze dwa znaki kodują symbol województwa (00, 02, ..., 32), dwa kolejne znaki kodują symbol powiatu (00, 01, ...99) oraz trzy ostatnie znaki kodują symbol gminy (000, ...).
Nazwa	Pełna polska nazwa danej gminy.
Rok	Rok w którym rejestrowano dane zmiany.
Wartość	Wartość obserwowanej zmiany.

3.7 Dotacje Unijne w Polsce

Wykorzystano zbiór danych na temat dotacji unijnych w Polsce w celu przeanalizowania wpływu tych dotacji na prędkość rozwoju ekonomicznego gmin i powiatów.

4 Model hurtowni danych

Przy budowaniu hurtowni zdecydowaliśmy się na użycie schematu gwiazdy. Główną przyczyną tego podejścia był fakt, iż nasze dane są relatywnie proste i nie ma potrzeby tworzenia zbyt skomplikowanych hierarchii. Nasz układ zawiera ciężką tabelę faktową, będącą opisem rozwoju danej gminy oraz dwa dodatkowe wymiary jakimi są tabela Region zawierająca dane użyteczne przy agregacji względem większej mniejszej ziarnistości terytorialnej (województwa i powiaty) oraz tabela Year zawierająca informacje o roku z którego pochodzą dane, informacje o partii politycznej wówczas rządzącej, imię oraz nazwisko osoby pełniącej stanowisko prezydenta w Polsce, wskaźnik inflacji oraz flagę określającą, czy Polska w tym roku należała do Unii Eurpejskiej.



Rysunek 2: Model fizyczny hurtowni

Kluczowymi miarami w hurtowni w tabeli **FactEconomicDevelopment** są:

- income_per_capita_start - dochód danej gminy w przeliczeniu na jednego mieszkańca na początku roku kalendarzowego
- income_per_capita_end - dochód danej gminy w przeliczeniu na jednego mieszkańca na końcu roku kalendarzowego
- income_per_capita_absolute - absolutny dochód danej gminy w danym roku kalendarzowym w przeliczeniu na 1 mieszkańca (dochód uzyskany na końcu roku kalendarzowego - dochód uzyskany na początku roku kalendarzowego)
- income_per_capita_relative - relatywny dochód danej gminy w danym roku kalendarzowym w przeliczeniu na 1 mieszkańca [(dochód uzyskany na końcu roku kalendarzowego - dochód uzyskany na początku roku kalendarzowego)/(dochód na początku roku kalendarzowego)]
- number_of_new_apartments - liczba lokali mieszkaniowych oddanych do użytku w danej gminie w danym roku kalendarzowym
- expenses_per_capita - wydatki gminy w przeliczeniu na jednego mieszkańca w danym roku kalendarzowym
- number_of_libraries - liczba bibliotek publicznych w danej gminie w danym roku kalendarzowym
- unemployment_male - wskaźnik bezrobocia wsród mężczyzn w danej gminie, w danym roku kalendarzowym
- unemployment_female - wskaźnik bezrobocia wsród kobiet w danej gminie, w danym roku kalendarzowym

Ponadto w tabeli **FactEUDonation** mamy miarę "amount" odnoszącą się do wysokości dotacji unijnej w danym regionie.

Przechodząc do kluczowych atrybutów w modelu, zacznijmy od tabeli **DimRegion**. Tabela ta zawiera nastepujące atrybuty:

- region_id - id danego regionu, jest to klucz podstawowy w tej tabeli. Jego odpowiednikiem w tabeli faktowej jest region_id będący kluczem obcym
- municipality - jest to pełna nazwa gminy
- county - jest to pełna nazwa powiatu do którego należy dana gmina
- voivodeship - jest to pełna nazwa województwa do którego należy dana gmina

- valid_from, valid_to - kolumny określające przedział czasowy, w którym dane były ważne

W tabeli **DimGovernmentProfile** mamy następujące atrybuty:

- ruling_party - partia polityczna, która rządziła w Polsce w danym roku
- president_name - imię oraz nazwisko prezydenta w danym roku
- prime_minister_name - imię oraz nazwisko premiera w danym roku

W tabeli **DimEconomicProfile** mamy następujące atrybuty

- unemployment_rate - wskaźnik bezrobocia w całym państwie
- inflation - poziom inflacji
- target_inflation_flag - flaga oznajmiająca czy inflacja jest na odpowiednim poziomie (1.5%-2.5%)
- deflation_flag - flaga oznajmiająca czy miała miejsce deflacja
- EU_flag - flaga oznajmaijąca czy Polska była członkiem Unii Europejskiej

W tabeli **DimDate** mamy następujące atrybuty

- date - konkretna data
- year - rok
- month - nazwa miesiąca
- week_of_year - numer tygodnia w skali roku
- day_of_week - nazwa dnia
- quarter - numer kwartału

5 Opis Procesu ETL

Po uzyskaniu danych z BDL GUS są one poddawane procesowi ETL. W tym kroku dane są pobierane ze źródła (w naszym przypadku pliku płaskiego), a następnie transformowane. Wszystkie dane dotyczące rozwoju gmin w Polsce, które zostały przez nas pobrane (łącznie 6 oddzielnych plików csv) zostały lekko zmodyfikowane i połączone w jeden plik csv. Zostało to w ten sposób zrobione gdyż miały stanowić one pierwsze źródło danych wykorzystywane przez nas w projekcie. Drugim źródłem danych natomiast były dane odnośnie dotacji unijnych znajdujące się w oddzielnym pliku csv.

5.1 DimDate

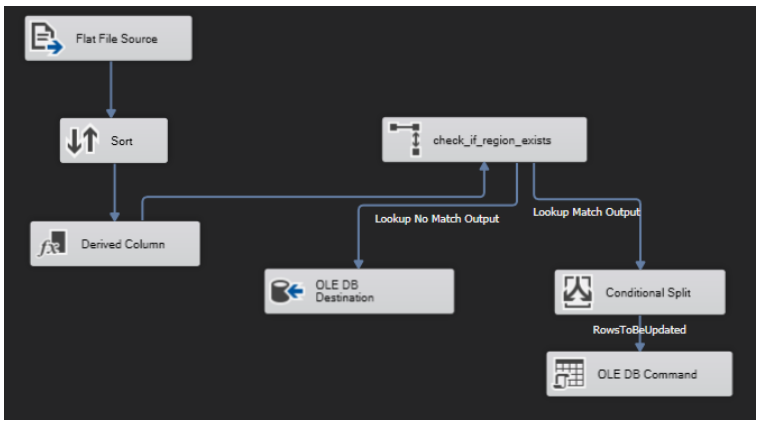
W pierwszej kolejności załadowaliśmy tabelę DimDate. Jest to tabela statyczna i jest ona załadowana jednorazowo. W tym celu napisaliśmy skrypt SQL ładujący dane do tej tabeli w odpowiednim dla nas formacie. Kod do załadowania wymiaru daty:

```
DECLARE @startDate DATE = '2000-01-01';
DECLARE @endDate DATE = '2300-12-31';
DECLARE @currentDate DATE = @startDate;

-- Tworzenie tabeli tymczasowej
CREATE TABLE #tempDates
(
    calendar_date DATE
);

-- Generowanie dat i zapisywanie ich do tabeli tymczasowej
WHILE @currentDate <= @endDate
BEGIN
    INSERT INTO #tempDates (calendar_date)
```

```
    VALUES (@currentDate);

    SET @currentDate = DATEADD(DAY, 1, @currentDate);
END

-- Wstawianie danych z tabeli tymczasowej do DimDate
INSERT INTO DimDate (date_id, calendar_date, week_of_year, quarter,
    year, month, day_of_week)
SELECT
    YEAR(calendar_date) * 10000 + MONTH(calendar_date) * 100 +
        DAY(calendar_date) AS date_id,
    calendar_date,
    DATEPART(ISO_WEEK, calendar_date) AS week_of_year,
    DATEPART(QUARTER, calendar_date) AS quarter,
    YEAR(calendar_date) AS year,
    CASE DATENAME(MONTH, calendar_date)
        WHEN 'January' THEN 'Styczeń'
        WHEN 'February' THEN 'Luty'
        WHEN 'March' THEN 'Marzec'
        WHEN 'April' THEN 'Kwiecień'
        WHEN 'May' THEN 'Maj'
        WHEN 'June' THEN 'Czerwiec'
        WHEN 'July' THEN 'Lipiec'
        WHEN 'August' THEN 'Sierpień'
        WHEN 'September' THEN 'Wrzesień'
        WHEN 'October' THEN 'Październik'
        WHEN 'November' THEN 'Listopad'
        WHEN 'December' THEN 'Grudzień'
    END AS month,
    CASE DATENAME(WEEKDAY, calendar_date)
        WHEN 'Sunday' THEN 'Niedziela'
        WHEN 'Monday' THEN 'Poniedziałek'
        WHEN 'Tuesday' THEN 'Wtorek'
        WHEN 'Wednesday' THEN 'Środa'
        WHEN 'Thursday' THEN 'Czwartek'
        WHEN 'Friday' THEN 'Piątek'
        WHEN 'Saturday' THEN 'Sobota'
    END AS day_of_week
FROM #tempDates;

-- Usuwanie tabeli tymczasowej
DROP TABLE #tempDates;
```

5.2 DimRegion

Przechodząc do właściwego procesu ETL, zacznijmy od Data Flow pod nazwą LoadRegionDim. Jest to pipeline do załadowania tabeli wymiarowej przetrzymującej informacje o regionach geograficznych w Polsce dla których dane zostały pobrane.



Rysunek 3: Pipeline do załadowania wymiaru regionu

Dane są ładowane z pliku płaskiego, następnie jest wykonywana operacja sortowania aby zatrzymać tylko wartości unikalne (operacja ta jest wykonywana na podstawie kolumny `region\_id` będącej unikalnym deskryptorem danego regionu. W dalszej kolejności dodawana jest kolumna `\valid_from` oraz `valid\_to`. Następnie sprawdzane jest czy dany region już istnieje w hurtowni. Jeśli nie, jest on dodawany, jeśli nie, jest on aktualizowany (jeśli wymaga aktualizacji)

5.2.1 Testy

Z uwagi na fakt, iż GUS udostępnia dane dla tych samych regionów co roku nie ma potrzeby aktualizacji już istniejących rekordów. Dane ładowane są jednorazowo, jednakże warto sprawdzić, czy liczba załadowanych regionów pokrywa się z faktyczną liczbą dla których określone są nasze dane. Opis testu:

- **cel**: testowane jest poprawne załadowanie wszystkich dostępnych regionów
- **sposób**: weryfikacja liczności wierszy w bazie danych oraz hurtowni oraz weryfikacja, czy dwie charakterystyczne obserwacje powtarzają się w obydwu przypadkach (MAX powiat oraz MIN gmina)
- **oczekiwany wynik**: oczekiwana liczba obserwacji to 2327, MAX powiat to powiat żywiecki, MIN gmina to Adamówka (2)

Results | Messages

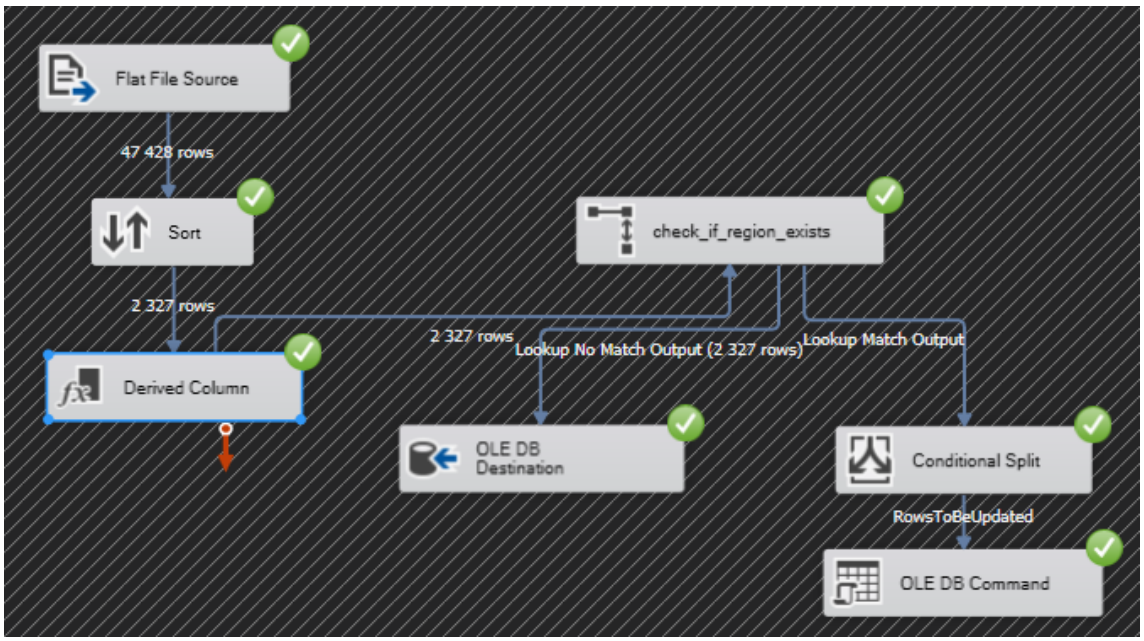
	(No column name)	(No column name)	(No column name)	(No column name)	(No column name)
1	Regions	Source	2327	Powiat żywiecki	Adamówka (2)
2	Regions	Target	2327	Powiat żywiecki	Adamówka (2)

Rysunek 4: Weryfikacja przeprowadzonego ETL

Dane zostały załadowane pomyślnie. W celu uzyskania powyższego sprawdzenia zostało wywołane następujące query.

```
Select 'Regions', 'Source', COUNT(DISTINCT Kod), MAX(Powiat),
    MIN(Gmina) FROM test_database.dbo.economic_development
UNION ALL
SELECT 'Regions', 'Target', COUNT(*), MAX(county), MIN(municipality)
    FROM dwh_project.dbo.DimRegion
```

Podczas przeprowadzonego ETL dla tej tabeli widać również, że ewidentnie zostały wybrane tylko unikalne wartości analizowanych regionów, co pokrywa się z wyżej pokazanym testem.



Rysunek 5: Data flow przeprowadzone pomyślnie

5.3 DimGovernmentProfile

W dalszej części załadowano tabelę wymiaru DimGovernmentProfile. Proces ten jest analogiczny do tabeli DimRegion. Docelowo dane byłyby scrapowane z internetu jednak w naszym rozwiązaniu przygotowaliśmy po prostu plik danych z odpowiednimi danymi.

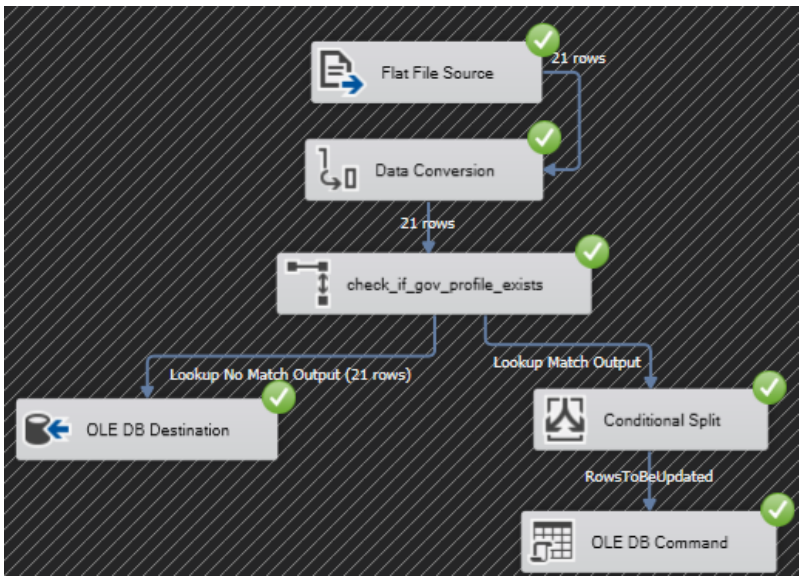
5.3.1 Testy

W podobny sposób testowane jest poprawne załadowanie naszych danych.

- **cel**: testowane jest poprawne załadowanie wszystkich dostępnych regionów
- **sposób**: weryfikacja liczności wierszy w bazie danych oraz hurtowni oraz weryfikacja, czy dwie charakterystyczne obserwacje powtarzają się w obydwu przypadkach
- **oczekiwany wynik**: oczekiwana liczba obserwacji to 21, MAX Partia to SLD, MIN Prezydent to Aleksander Kwaśniewski

	(No column name)	(No column name)	(No column name)	(No column name)	(No column name)
1	GovProfile	Source	21	'Sojusz Lewicy Demokratycznej (SLD)'	'Aleksander Kwaśniewski'
2	GovProfile	Target	21	'Sojusz Lewicy Demokratycznej (SLD)'	'Aleksander Kwaśniewski'

Rysunek 6: Weryfikacja przeprowadzonego ETL



Rysunek 7: Data flow przeprowadzone pomyślnie

5.4 DimEconomicProfile

Ostanią tabelą wymiaru jest tabela DimEconomicProfile zawierającą uproszczoną sytuację ekonomiczną w Polsce w danym roku. Podobnie jak dla DimGovernmentProfile te dane docelowo byłyby scrapowane z internetu.

5.4.1 Testy

W podobny sposób testowane jest poprawne załadowanie naszych danych.

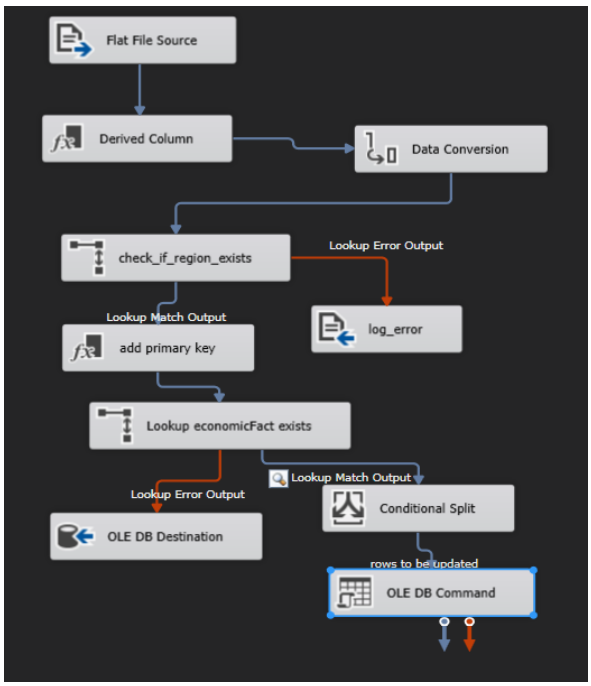
- **cel**: testowane jest poprawne załadowanie wszystkich dostępnych regionów
- **sposób**: weryfikacja liczności wierszy w bazie danych oraz hurtowni oraz weryfikacja, czy dwie charakterystyczne obserwacje powtarzają się w obydwu przypadkach
- **oczekiwany wynik**: oczekiwana liczba obserwacji to 21, MAX inflacja to 14.4, MIN bezrobocie to 5.5

	(No column name)	(No column name)	(No column name)	(No column name)	(No column name)
1	EconProfile	Source	21	14,4	5,5
2	EconProfile	Target	21	14,4	5,5

Rysunek 8: Weryfikacja przeprowadzonego ETL

5.5 FactEconomicDevelopment

Przechodząc do Data Flow dla tabeli faktowej FactEconomicDevelopment całość prezentuje się następująco:



Rysunek 9: Pipeline do załadowania faktu EconomicDevelopment

Dane są ładowane z pliku płaskiego, nastepnie wyliczne są nowe kolumny takie jak income_per_capita_absolute będącą całkowitą zmianą dochodu per capita w przeciągu roku kalendarzowego, income_per_capita_relative będąca procentową zmianą w dochodzie per capita oraz date_id będąca kluczem obcym, niezbędnym do powiązania tabeli faktowej z wymiarem daty. Następnie dokonywana jest konwersja typów kodu regionu na wartość numeryczną aby wykonać operację lookup. W operacji lookup sprawdzane jest, czy region o podanym kodzie już istnieje (powinien istnieć gdyż GUS udostępnia dane dla tych samych regionów). Jeśli taki kod nie istnieje rejestrowany jest error log w specjalnie do tego stworzonym pliku. Jeśli taki region istnieje dodawany jest unikatowy klucz główny w tej tabeli będący sklejeniem kodu regionu oraz klucza obcego daty (date_id). W dalszej części znowu wywoływany jest lookup, tym razem do tabeli faktowej w celu weryfikacji czy taki wiersz już w naszej bazie nie istnieje. Jeśli istnieje to znaczy, że należy wprowadzić korektę i w tym celu wywoływana jest komenda UPDATE aby zaktualizować rekord z nowymi parametrami. W przeciwnym wypadku nowe rekordy ładowane są do hurtowni.

5.5.1 Testy

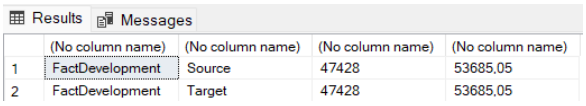
W celu sprawdzenia poprawności przeprowadzonego procesu ETL sprawdzono liczbę rekordów oraz największą wartość dochodu per capita w tabeli faktowej z hurtowni danych oraz tabeli z pomocniczej bazy danych, do której zostały załadowane dane z pliku płaskiego.
Opis testu:

- **cel**: testowane jest poprawne załadowanie wszystkich dostępnych danych faktowych
- **sposób**: weryfikacja liczności wierszy w bazie danych oraz hurtowni oraz weryfikacja, czy charakterystyczne obserwacje powtarzają się w obydwu przypadkach (największą wartość dochodu per capita)
- **oczekiwany wynik**: oczekiwana liczba obserwacji to 47428, największą wartość dochodu per capita to 53685.05

W celu weryfikacji testu wywołano następujące query w SQL:

```
Select 'FactDevelopment', 'Source', COUNT(*), MAX(Dochod) FROM
    test_database.dbo.economic_development
UNION ALL
SELECT 'FactDevelopment', 'Target', COUNT(*),
    MAX(income_per_capita_start) FROM
    dwh_project.dbo.FactEconomicDevelopment
```

w skutek czego dostano wynik zapytania:



Results Messages

	(No column name)	(No column name)	(No column name)	(No column name)
1	FactDevelopment	Source	47428	53685,05
2	FactDevelopment	Target	47428	53685,05

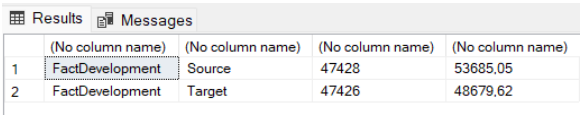
Rysunek 10: Wynik testu poprawności załadowanych danych

Zatem test został zaliczony pomyślnie.

Kolejnym testem było usunięcie z tabeli z hurtowni dwóch obserwacji z największą wartością "income_per_capita_start" i ponowne uruchomienie data flow, aby załadować brakujące dane do tabeli.
Opis testu:

- **cel**: testowane jest poprawne załadowanie nowych danych (w tym wypadku dwóch usuniętych) oraz aktualizację istniejących rekordów.
- **sposób**: weryfikacja liczności wierszy w bazie danych i hurtowni, weryfikacja, czy wszystkie charakterystyczne obserwacje powtarzają się w obydwu przypadkach. W tym celu usunięto dwie obserwacje o najwyższych wartościach pola "income_per_capita_start".

```
DELETE FROM dwh_project.dbo.FactEconomicDevelopment
WHERE income_per_capita_start IN (
    SELECT TOP 2 income_per_capita_start
    FROM dwh_project.dbo.FactEconomicDevelopment
    ORDER BY income_per_capita_start DESC
);
```



Results Messages

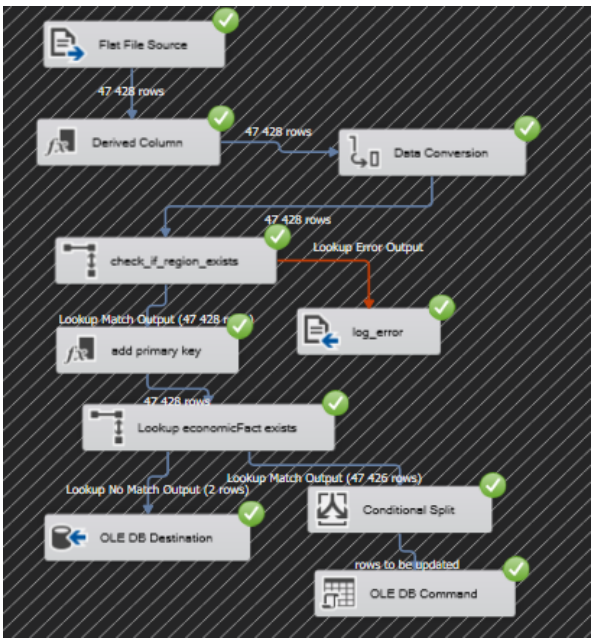
	(No column name)	(No column name)	(No column name)	(No column name)
1	FactDevelopment	Source	47428	53685,05
2	FactDevelopment	Target	47426	48679,62

Rysunek 11: Weryfikacja usunięcia z hurtowni dwóch rekordów o najwyższych wartościach pola "income"

- **oczekiwany wynik**: oczekiwana liczba obserwacji to 47428, największą wartość dochodu per capita to 53685.05

Weryfikacja oczekiwanego wyniku:

- Wywołanie data flow w celu wstawienia brakujących obserwacji.



Rysunek 12: Wywołanie data flow. Dwa wierwsze są dodawane do hurtowni

- Weryfikacja poprawności załadowanych danych.

Results | Messages

	(No column name)	(No column name)	(No column name)	(No column name)
1	FactDevelopment	Source	47428	53685,05
2	FactDevelopment	Target	47428	53685,05

Rysunek 13: Weryfikacja poprawności załadowanych danych

Jak widać proces ładowania danych przebiegł pomyślnie i dane zostały załadowane w sposób poprawny.

Przejdźmy do sprawdzenia czy dane zostaną w poprawny sposób zaktualizowany jeśli dojdzie do korekty kluczowych miarek.
Opis testu:

- **cel**: testowane jest aktualizacja zmienionych obserwacji w pliku źródłowym
- **sposób**: weryfikacja liczności wierszy w bazie danych oraz hurtowni oraz weryfikacja, czy charakterystyczne obserwacje powtarzają się w obydwu przypadkach (największą wartość bezrobotnych mężczyzn), a przede wszystkim wywołanie komendy select i zobaczenie, czy obserwacja została zaktualizowana
- **oczekiwany wynik**: oczekiwana liczba obserwacji to 47428, największą wartość liczby bezrobotnych mężczyzn to 32726. Jeśli chodzi o pierwotną wartość dla obszaru o region_id = 201011 oraz wartości pola year w DimDate równym 2002, pola niezatrudnionych mężczyzn wynosiła ona -1, gdyż tak kodujemy braki danych (dane zawierające informację o bezrobociu zaczynają się od roku 2003). Po aktualizacji natomiast oczekujemy wartości 12, gdyż właśnie na taką wartość w celach testowych zostało podmienione te pole w pliku źródłowym.

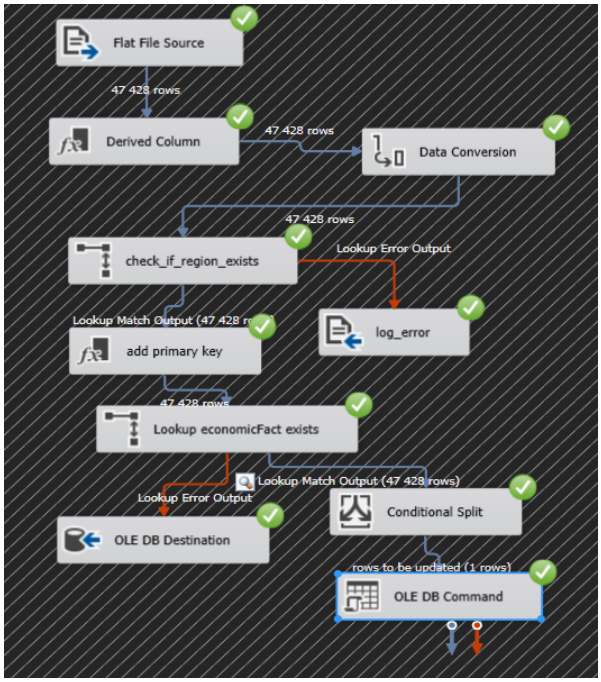
Weryfikacja testu.
Początkowo nasze dane wyglądały następująco:

Results Messages

	region_id	unemployment_male	year
1	201011	-1	2002
2	201022	-1	2002
3	201032	-1	2002
4	201043	-1	2002
5	201052	-1	2002
6	201062	-1	2002
7	3218032	-1	2002
8	3218043	-1	2002
9	3218053	-1	2002
10	3261011	-1	2002
11	3262011	-1	2002
12	3263011	-1	2002
13	3217023	-1	2002
14	3217033	-1	2002

Rysunek 14: Pierwotna wersja danych

Teraz w pliku płaskim podmieńmy wartość bezrobocia wśród mężczyzn na 12 i wywołajmy data flow jeszcze raz.



Rysunek 15: Data flow na zmodyfikowanych danych

Zweryfikujmy nowe dane:

Results Messages

	region_id	unemployment_male	year
1	201011	12	2002
2	201022	-1	2002
3	201032	-1	2002
4	201043	-1	2002
5	201052	-1	2002
6	201062	-1	2002
7	3218032	-1	2002
8	3218043	-1	2002
9	3218053	-1	2002
10	3261011	-1	2002
11	3262011	-1	2002
12	3263011	-1	2002
13	3217023	-1	2002
14	3217033	-1	2002
15	3217043	-1	2002
16	3217052	-1	2002

Rysunek 16: Wynik zapytania SQL z tabeli FactEconomicDevelopment

Operacja aktualizacji przeszła pomyślnie. Warto jednak sprawdzić, czy nie zmieniła się liczba

wierszy oraz weźmy maksymalną liczbę bezrobotnych mężczyzn i zobaczmy, czy wyniki są zgodne z prawdziwymi danymi.

Results | Messages

	(No column name)	(No column name)	num_rows	max_unemployed_male
1	FactDevelopment	Source	47428	32726
2	FactDevelopment	Target	47428	32726

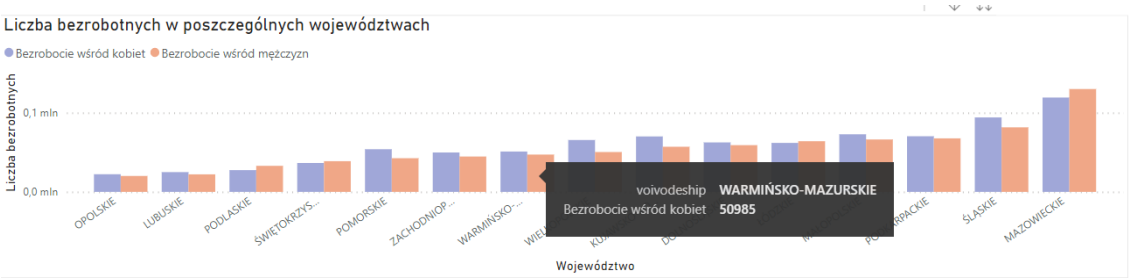
Rysunek 17: Weryfikacja liczności wierszy w tabeli źródłowej oraz faktowej

Liczność wierszy oraz maksymalna liczba mężczyzn bezrobotnych się zgadza, można więc przypuszczać, iż aktualizacj przebiegła pomyślnie.

Na sam koniec sprawdzimy poprawność wizualizacji w PowerBI.
Opis testu:

- **cel**: testowana jest poprawność przedstawionych wizualizacji
- **sposób**: weryfikacja losowo wybranych wyników przedstawionych na wykresie, sprawdzając i porównując to z otrzymanymi wynikami na skutek wywołania query w SQL w naszej hurtowni, odzwierciedlającego oczekiwane agregacje wykonywane pod spodem w PowerBI
- **oczekiwany wynik**: oczekiwane wyniki to te przedstawione zawsze na wizualizacjach. Jeśli wartości uzyskane w formie tabelarycznej jako wynik zapytania *SELECT* pokrywają się z danymi na wykresie, oznacza to że wizualizacja jest przedstawiona poprawnie.

Weryfikacja opisanego testu:



Rysunek 18: Liczba kobiet bezrobotnych w Warmińsko-Mazurskim w roku 2014, dane z raportu

Oczekiwaną zatem wartością jest 50985 wśród bezrobotnych kobiet w województwie Warmińsko-Mazurskim w roku 2014.
Przejdźmy do weryfikacji tego wyniku.

```
SELECT voivodeship, SUM(unemployment_male) AS sum_male,
    SUM(unemployment_female) AS sum_female
FROM FactEconomicDevelopment JOIN
DimRegion on FactEconomicDevelopment.region_id = DimRegion.region_id
WHERE date_id = 20140101
GROUP BY voivodeship;
```

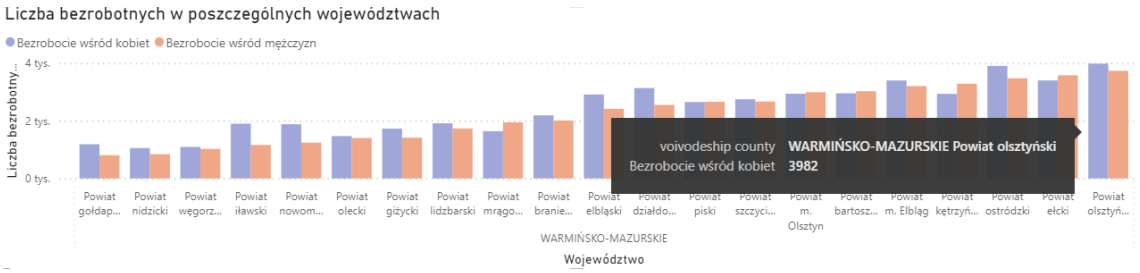
W celu sprawdzenia poprawności wywołano powyższe query. Wynik zapytania poniżej:

Results | Messages

	voivodeship	sum_male	sum_female
1	MAŁOPOLSKIE	66227	72800
2	WARMIŃSKO-MAZURSKIE	47154	50985
3	ŁÓDZKIE	64029	61895
4	ŚWIĘTOKRZYSKIE	38763	36512
5	KUJAWSKO-POMORSKIE	57033	70078
6	PODKARPACKIE	67627	70305
7	PODLASKIE	32916	27478
8	DOLNOŚLĄSKIE	59044	62518
9	POMORSKIE	42542	53912
10	MAZOWIECKIE	130009	119263
11	WIELKOPOLSKIE	50342	65542
12	ŚLĄSKIE	81574	94101
13	LUBUSKIE	22135	24980
14	ZACHODNIOPOMORSKIE	44670	49795
15	OPOLSKIE	20092	22269

Rysunek 19: Weryfikacja informacji przy użyciu SQL query

Wartość się zgadza, więc można przypuszczać, że wizualizacja działa poprawnie. Zejdźmy jednak jeszcze niżej w hierarchię i również zweryfikujmy działanie.



Rysunek 20: Liczba kobiet bezrobotnych w powiecie olsztyńskim w roku 2014, dane z raportu

Oczekiwaną zatem wartością jest 3982 wśród bezrobotnych kobiet w województwie Warmińsko-Mazurskim, w powiecie olsztyńskim w roku 2014.

```
SELECT  county ,SUM(unemployment_male) AS sum_male,
    SUM(unemployment_female) AS sum_female
FROM FactEconomicDevelopment JOIN
DimRegion on FactEconomicDevelopment.region_id = DimRegion.region_id
WHERE date_id = 20140101 and voivodeship='WARMIŃSKO-MAZURSKIE'
GROUP BY county;
```

W celu sprawdzenia poprawności wywołano powyższe query. Wynik zapytania poniżej:

Results | Messages

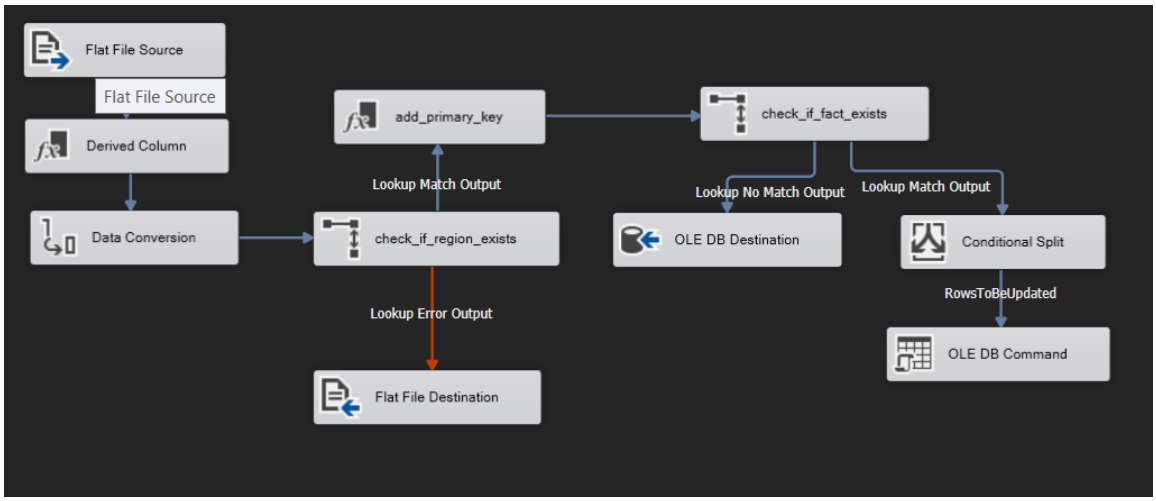
	county	sum_male	sum_female
6	Powiat giżycki	1413	1727
7	Powiat gołdapski	806	1185
8	Powiat iławski	1163	1899
9	Powiat kętrzyński	3284	2932
10	Powiat lidzbarski	1734	1912
11	Powiat m. Elbląg	3202	3396
12	Powiat m. Olsztyn	2991	2938
13	Powiat mrągowski	1944	1639
14	Powiat nidzicki	843	1053
15	Powiat nowomiejski	1242	1880
16	Powiat olecki	1403	1468
17	Powiat olsztyński	3732	3982
18	Powiat ostródzki	3474	3902
19	Powiat piski	2657	2648
20	Powiat szczycieński	2667	2746
21	Powiat węgorzewski	1026	1094

Rysunek 21: Weryfikacja informacji przy użyciu SQL query

Tutaj również wartości się zgadzają. Testowanie zatem przeszło pomyślnie.

5.6 FactEUDonation

Na koniec opiszemy ładowanie danych do tabeli FactEUDonation. Należy zaznaczyć, że dane zostały przez nas wygenerowane, a nie sciągnięte z oficjalnego źródła. Postaraliśmy się, aby ich agregacja możliwie dobrze pokrywała się z rzeczywistością (aby raport wyglądał rozsądnie, a nie był tylko szumem) jednak oczywiście nie jest to idealna odwzorowanie.



Rysunek 22: Pipeline do załadowania faktu EUDonation

5.6.1 Testy

Opis testu:

- **cel**: testowane jest poprawne załadowanie wszystkich dostępnych danych faktowych
- **sposób**: weryfikacja liczności wierszy w bazie danych oraz hurtowni oraz weryfikacja, czy charakterystyczne obserwacje powtarzają się w obydwu przypadkach (największa wartość dotacji)
- **oczekiwany wynik**: oczekiwana liczba obserwacji to 44213, największą wartość dotacji to 474886515.

W celu weryfikacji testu wywołano następujące query w SQL:

```
Select 'FactDonation', 'Source', COUNT(*), MAX(Dotacja) FROM
    test_database.dbo.eu_funds
UNION ALL
SELECT 'FactDonation', 'Target', COUNT(*), MAX(amount) FROM
    dwh_project.dbo.FactEUDonation
```

	(No column name)	(No column name)	(No column name)	(No column name)
1	FactDonation	Source	44213	474886515,00
2	FactDonation	Target	44213	474886515,00

Rysunek 23: Weryfikacja liczności wierszy w tabeli źródłowej oraz faktowej

Wartości dla obu tabel są takie same.

Kolejny test dla tej tabeli faktowej ma na celu sprawdzić poprawność ładowania nowych danych oraz aktualizację istniejących danych.

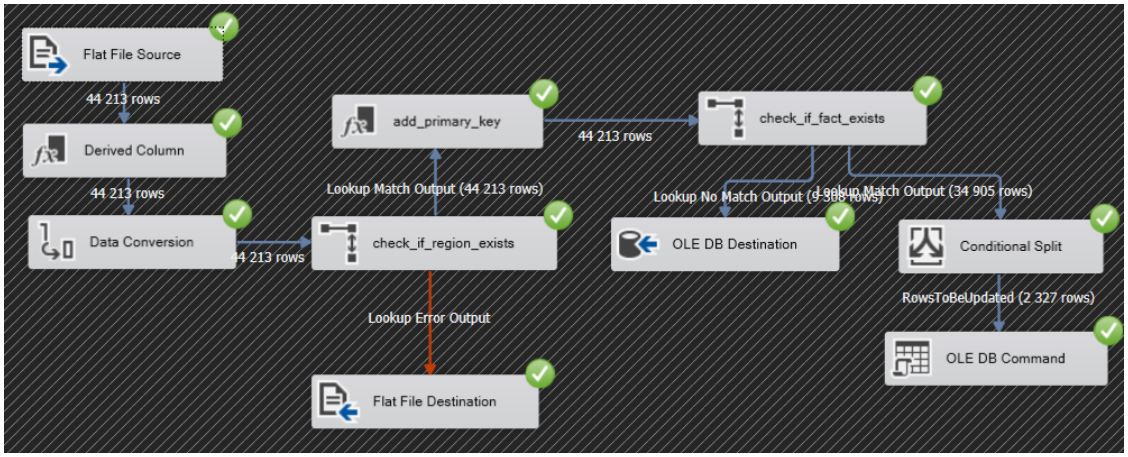
- **cel**: testowane jest poprawne ładowanie nowych i aktualizowanie starych danych w hurtowni
- **sposób**: usunięcie i modyfikacja istniejących danych w hurtowni oraz ponowne uruchomienie procesu ETL. Weryfikacja poprawności działania procesu oraz danych.

  Usuniemy wszystkie obserwacje po roku 2018 oraz zasymulujemy brak danych w rubryce Dotacja dla danych z przed 2005.

- **oczekiwany wynik**: oczekiwana liczba obserwacji to 44213, średnia wartość dotacji to 16529159.0686.

Usunięcie oraz modyfikacja rekordów została wykonana za pomocą polecenia

```
DELETE F FROM dwh_project.dbo.FactEUDonation F JOIN
dwh_project.dbo.DimDate D on D.date_id=F.date_id
WHERE D.year > 2018

UPDATE F
SET
    amount = -1
FROM dwh_project.dbo.FactEUDonation F JOIN dwh_project.dbo.DimDate D
    on F.date_id = D.date_id
WHERE D.year < 2005
```



Rysunek 24: Data flow na usuniętych i zmodyfikowanych danych

Proces ETL działa bez problemowo. Istniejące, niezmienione obserwacje ignoruje dodając tylko nowe oraz aktualizując stare. Liczba wierszy dodanych (9308) oraz zaktualizowanych (2327) pokrywa się z wynikiem wykonanego zapytania SQL.

6 Opis planowanych raportów dla użytkowników

Dołączając narzędzie z zakresu Business Intelligence planujemy raportowanie z możliwością agregacji na poziomie województw, powiatów oraz gmin. Chcielibyśmy przedstawić analizy uwzględniające zmianę dochodów oraz wydatków gmin na przestrzeni lat oraz zestawienie to z aktualnie rządzącymi partiami w celu analizy, czy ma to faktyczny wpływ.

Posiadając dane od 2002 roku jesteśmy również w stanie zweryfikować, sytuację gospodarczą przed przystąpieniem Polski do Unii Europejskiej oraz po przystąpieniu co jest ciekawym obszarem do analizy.

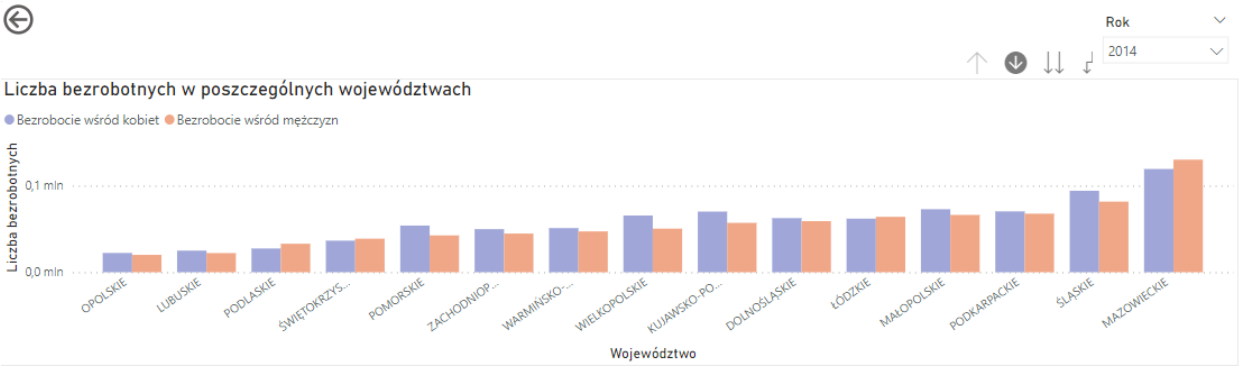
Chcielibyśmy również zweryfikować jak zmienia się stopa bezrobocia w Polsce w różnych obszarach administracyjnych na przestrzeni lat oraz czy wskaźnik inflacji wpływał na wydatki generowane przez gminy, liczbę nowych jednostek mieszkalnych oraz na dochód generowany per capita.

Na sam koniec chcielibyśmy zestawić gminy/powiaty/województwa względem najważniejszych miar aby wyłonić najbardziej rozwijające się w danych latach obszary oraz zweryfikować jak wygląda ta tendencja (w których latach możemy obserwować przestój gospodarczy/fazę prężnego rozwoju).

7 Przykładowe wizualizacje

Raportowania zostały stworzone w środowisku PowerBI for Desktop. Dane znajdujące w hurtowni pozwoliły stworzyć niezbędne wizualizacje do analizy rozwoju jednostek aglomeracyjnych w naszym kraju.

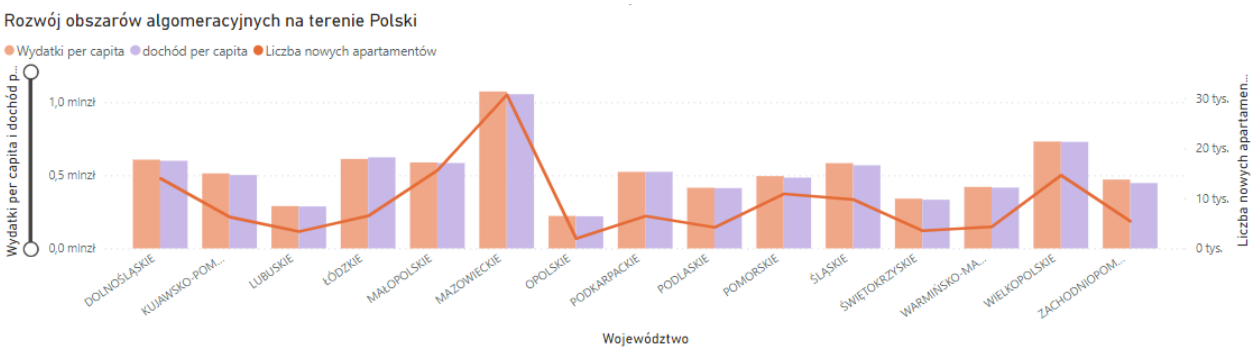
Chęć analizy rozwoju poszczególnych obszarów w Polsce skłoniła nas do przebadania między innymi liczności bezrobocia na terenie województw/powiatów/gmin.



Rysunek 25: Bezrobocie w danych jednostkach algomeracyjnych

Wykres umożliwa zmianę roku, oraz przechodzenie wgłąb hierachi regionów, co pozwala zmieniać ziarnistość danych od najbardziej ogólnych - województwa, przez trochę bardziej szczegółowe - powiaty, do najbardziej szczegółowych - gminy.

Kolejną przygotowaną wizualizacją jest ta przedstawiająca zmianę w dochodach, wydatkach przypadających na jedną osobę oraz oddanych nowych apartamentach do użytkowania. Wykres również pozwala zmieniać ziarnistość danych względem jednostek aglomeracyjnych.



Rysunek 26: Rozwój w danych jednostkach algomeracyjnych

Na wykresie słupkowym przedstawione są dochody oraz wydatki przypadające na jednego mieszkańca, natomiast wykres liniowy pokazuje ilość lokali użytkowych oddanych w danym roku kalendarzowym. Element umożliwiający zmianę roku w którym badamy poszczególne charakterystyki to fragmentor umieszczony powyżej pierwszej wizualizacji. Dane są zatem zsynchronizowane i wizualizacje automatycznie się odświeżają po zmianie roku w którym chcemy dokonywać analizy. Forma agregacji miarek na tej stronie raportowej ustawiona jest na sumę.

Na kolejnej stronie w przygotowanym raporcie, możemy odnotować zestawienie średniego dochodu-/wydatków przypadających na jednego obywatela oraz stopę bezrobocia w zależności od panującej w Polsce partii politycznej. Tym razem agregacja miarek została ustawiona na średnią, gdyż jest to lepsza forma agregacji z uwagi na fakt, że część partii rządziła więcej lat od pozostałych i suma nie byłaby tu adekwatna. Tutaj również została zaprezentowana hierarchia składająca się z partii politycznej/prezydenta/ministra. Wizualizacje w efektywny sposób pozwalają analizować wpływ władz rządzących na rozwój regionów na terenie Polski.

Rysunek 27: Rozwój w danych jednostkach algomeracyjnych w zestawieniu z partiami rządzącymi

Oprócz wizualizacji została również zastosowana inteligentna narracja i choć nie działa ona idealnie w języku Polskim, to zawiera wstępną analizę taką jak procentowe zmiany poszczególnych miarek w zależności od rządzącej partii.

8 Opis warstwy raportowej

W tej sekcji krótko opiszemy nasze działanie na danych wewnątrz PowerBI. Nasze wizualizacje nie wymagały dodatkowych miar, ponieważ wszelkie potrzebne miary pozyskaliśmy bezpośrednio ze źródła danych lub zostały one utworzone w procesie ETL. Zostały jednak dodane nowe relacje, które zaburzają model gwiazdy, dlatego nie zostały one dodane w hurtowni. Takimi relacjami są

- Relacja DimEconomicProfile-DimGovernmentProfile na podstawie roku
- Relacja FactEUDonation-DimGovernmentProfile na podstawie roku

W drugiej relacji tabela FactEUDonation nie posiada kolumny rok, ma ona jednak swoją przypisaną date na podstawie date_id. W naszej bazie klucz główny w wymiarze daty to po prostu skonkatenowany rok miesiąc i dzień miesiąca zatem transformacja tej kolumny bezpośrednio daje nam rok donacji bez potrzeby łączenia tabel.

Wymagane było od nas także ustawienie odpowiedniej lokalizacji (w sensie kulturowym) niektórych kolumn, tak aby przecinki w liczbach były odpowiednio traktowane jako separatory części całkowitej i ułamkowej.

Przygotowaliśmy także odpowiednie hierarchie regionu oraz rządu.

9 Podsumowanie

Uważamy, że przygotowane przez nas dane i wizualizacje są informatywne i pozwalają na wyciągnięcie ciekawych wniosków. Możliwość podziału danych według regionów pozwala na bardzo szczegółową analizę rozwoju danych obszarów.

Niestety ograniczniem naszej wizualizacji jest nieprzedstawienie danych w postaci mapy co znacznie ułatwiłoby analizę w skali całego państwa nawet na poziomie gmin. Jest to coś czym chcielibyśmy się zająć, gdybyśmy mogli poświęcić na całość więcej czasu.

Podobnie ogracznieniem jest symulacja danych funduszy unijnych. Pomimo starań, aby dane prezentowały się w sposób rozsądny i odwzorujące rzeczywistość, są one nadal nieprawdziwe. Z tego powodu nie można wyciągnąć z nich żadnych szczegółowych wniosków. Ponieważ agregacja danych na przestrzeni lat czy województw dość dobrze przypomina rzeczywiste dane to nadal możliwe jest stwierdzenie pewnych globalnych faktów.

Ostatecznie, pomimo swoich ograczneń przygotowane przez nas rozwiązanie zdaje się być dobrym narzędziem do analizy rozwoju ekonomicznego w polsce.

10 Podział pracy w zespole

Szymon Gut

- Zaplanowanie architektury hurtowni danych
- Dokumentacja
  - Korzyści z perspektywy odbiorcy
  - Sekcja pozyskane dane opisująca wykorzystywane zbiory danych (oprócz podsekcji dotyczącej dotacji unijnych)
  - opis procesu ETL wymiaru Daty, Regionu oraz GovernmentProfile, a także dla tabeli faktowej EconomicDevelopment
  - opis i przygotowanie testów dla tabeli Region oraz tabeli EconomicDevelopment
  - Planowane raportowanie
  - Przykładowe wizualizacje
- Przygotowanie procesu ETL dla tabel Region, Date oraz EconomicDevelopment
- Skrypt łączący mniejsze pliki .csv do jednego dużego pliku płaskiego wykorzystywanego w procesach ETL.
- Wizualizacje w PowerBI widoczne na pierwszej i drugiej stronie

Jan Krężel:

- Zaplanowanie architektury hurtowni danych
- Dokumentacja
  - Diagram proponowanej architektury danych
  - Sekcja pozyskane dane dotycząca dotacji unijnych
  - Opis procesu ETL dla wymiaru EconomicProfile, GovernmentProfile oraz dla tabeli faktowej dotacji unijnych
  - Opis i przygotowanie testów dla wyżej wymienionych tabel
  - Opis warstwy raportowej
  - Podsumowanie
- Przygotowanie procesu ETL dla tabel GovernmentProfile, EconomicProfile oraz EUDonation
- Rozbudowanie procesu ETL dla pozostałych tabel
- Przygotowanie symulowanych danych dotacji unijnych
- Wizualizacje w PowerBI widoczne na trzeciej i czwartej stronie.
- Prezentacja końcowa