Hurtownie danych i systemy Business Inteligence
Analiza danych zasięgu sieci telefonicznej w Polsce

Karol Degórski i Jakub Lis

14 czerwca 2022

1 Opis biznesowy

Celem projektu jest przeanalizowanie jakie czynniki mają wpływ na dostęp do zasięgu sieci telefonicznej w Polsce oraz identyfikacja potencjalnych obszarów, w których można postawić nowe maszty BTS. Przeanalizowaliśmy wpływ gęstości zaludnienia, struktury wiekowej, wykształcenia, bezrobocia oraz odestka osób posiadających telefon na liczbę masztów w danym regionie. Wykorzystaliśmy dane o masztach transmisyjnych pochodzące ze strony: http://beta.btsearch.pl. Zawierają one informację o operatorze, wykorzystywanych technologiach oraz lokalizacji danego masztu. Aby uzyskać dane ekonomiczne skorzystaliśmy z bazy danych GUS (Bank Danych Lokalnych). Połączone w ten sposób dane zwizualizowaliśmy za pomocą narzędzia klasy Business Intelligence.

Analiza wydaje się być kluczowa głównie z punktu widzenia operatorów telefonii komórkowej, ponieważ pozwala ona na znalezienie nowych grup użytkowników, którzy mogą być zainteresowani uzyskaniem dostępu do zasięgu. Dzięki temu zyski firm telekomunikacyjnych mogą zostać zwiększone. Ponadto dana firma może porównać się do konkurencji i sprawdzić jak jej zasięg wypada w porównaniu z innymi. Przeprowadzone przez nas analizy zapewniają również rzetelny dostęp do informacji dla zwykłych ludzi, aby mogli oni sprawdzić, czy mają dostęp do danej sieci oraz poszerzyć swoją wiedzę o ciekawe statystyki.

2 Opis danych

2.1 Dane dotyczące masztów sieciowych

Źródłem danych są wyeksportowane pliki CLF ze strony `http://beta.btsearch.pl/bts/` dotyczące masztów sieciowych. W danych znajdują się takie informacje, jak

- *LAT i LON* - współrzędne geograficzne masztu;
- *POS-RAT* - dokładność współrzędnych miejsca, wartością -1 oznaczone są dokładnie podane współrzędne;
- *DESC* - opis w postaci adresu i informacji na temat masztu, np. że jest na dachu budynku, albo że jest osobną wieżą; także informacja nt. operatora masztu (T-mobile, Orange, itd.);
- *SYS* - kod odpowiadający standardowi masztu, np. GSM, LTE, 5G;

Dokładniejszy opis tych zmiennych jest dostępny na stronie `https://sites.google.com/site/clfgmon/clf4?authuser=0`, ale w przypadku niewymienionych kolumn najczęściej dane zawierają wartość -1, oznaczającą brak informacji.

2.2 Dane ekonomiczne pochodzące z GUS

Drugim źródłem danych jest Bank Danych Lokalnych GUS. Zawiera on bardzo dużo różnych tabel dotyczących ludności Polski. Dane te są w postaci plików CSV. Wybraliśmy kilka najbardziej interesujących tabel, są to:

- Gęstość zaludnienia - ludność na $1km^2$ na poziomie gmin;
- Ludność według grup wieku i płci - dane na poziomie gmin;
- Bezrobocie zarejestrowani wg wieku - dane na poziomie powiatów;
- Ludność w wieku 13 lat i więcej wg grup wieku i poziomu wykształcenia - dane na poziomie powiatów;
- Wyposażenie w niektóre przedmioty trwałego użytkowania w % ogółu gospodarstw domowych - dane na poziomie województw;

Ponadto pobraliśmy z rejestru TERYT plik CSV dekodujący kod gminy na nazwę gminy, nazwę powiatu oraz nazwę powiatu.

Uzupełniającym źródłem danych jest strona `https://skuteczneraporty.pl`. Pobraliśmy z niej plik XLSX zawierający informację o kodach pocztowych należących do danych gmin. Informacja ta okazała się niezbędna do późniejszego procesu wizualizacji danych w narzędziu Business Intelligence.

3 Opis architektury

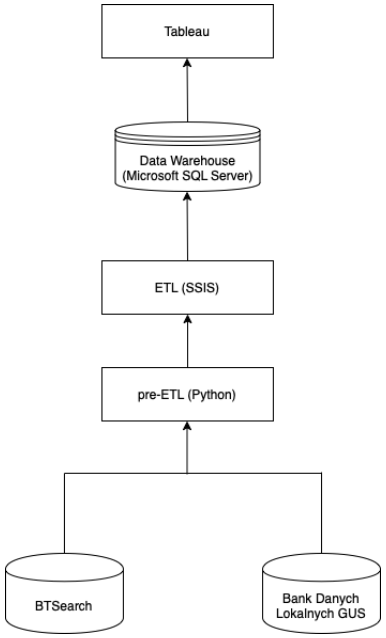
W niniejszym punkcie przedstawimy opis architektury hurtowni danych

3.1 Podział na warstwy

Architekrurę DWH/BI prezentujemy na rysunku 1. Najniższą warstwę stanowią dwie bazy danych: btsearch (pliki CLF) oraz Bank Danych Lokalnych GUS (pliki csv). Z uwagi na problemy z poprawnym działaniem funkcjonalności ETL, dane są wstępnie obrabiane z wykorzystaniem języka Python. Następnie dane są łączone oraz obrabiane poprzez proces ETL, który omówimy w kolejnym punkcie. Do przeprowadzenia tego procesu wykorzystaliśmy SQL Server Integration Services (SSIS). Dane powstałe w ten sposób zasilają hurtownię danych w modelu gwiazdy w Microsoft SQL Server. Ostatnią warstwą jest warstwa narzędzia klasy Business Inteligence, w naszym przypadku Tableau.

3.2 Wstępna obróbka danych

Etap ten miał za zadanie przygotować pliki uzyskane ze źródeł do właściwego procesu ETL. Początkowo nie planowaliśmy wykorzystywać żadnych innych narzędzi do obróbki danych poza procesem ETL. Jednakże w czasie prac okazało się, że nie wszystkie funkcje narzędzia SSIS działają



Rysunek 1: Architektura DWH/BI

poprawnie. Mieliśmy duże problemy z dokonaniem pivotu (denormalizacją tabeli). Ostatecznie postanowiliśmy wykorzystać Python wraz z biblioteką Pandas do wstępnej obróbki danych. Wykonaliśmy pivot tabel pochodzących z Gus-u, tak aby to wiersze zawierały informacje dla poszczególnych lat. Ponadto do plików CLF dodajemy id województw, do nazw, za pomocą kodowania znalezionego w bazie TERYT. Ujednolicamy również nazwy miasta Warszawa, ponieważ rozważamy miasto jako jedność, a nie w podziale na poszczególne dzielnice. Na tym etapie wykonaliśmy niezbędne minimum przekształceń danych. Główną część obróbki dokonujemy w procesie ETL, który omówimy w kolejnym punkcie.

3.3 Proces ETL

Proces ETL składa się z 3 kroków, z uwagi na konieczność zasilenia 3 tabel hurtowni danych. Omówimy teraz każdy z nich.

3.3.1 Załadowanie danych do tabeli faktów Maszty

Dane są wczytywanie z pliku płaskiego CLF. Rozdzielamy je na poszczególne kolumny, a błędne rekordy przekierowujemy do error loga. Zaczynamy od wyciągnięcia z danych potrzebnych informacji za pomocą szeregu przekształceń derived column. W danych zawarta jest kolumna DESC oznaczająca opis lokalizacji, w szczególności oprócz informacji o miejscowości pojawiają się dane, o tym gdzie znajduje się maszt, np. na dachu jakiegoś budynku, albo że jest to maszt własny. Pod naszą analizę chcemy tylko informacje o gminie masztu i ewentualnie o miejscowości. Rzeczywiście można to było wyciągnąć, ale potrzebne było dużo operacji derived column operujących na strin-

gach. Bez wchodzenia w szczegóły, dzielimy string, sprawdzamy - czy zawiera informacje o gminie i ją wyciągamy, a jeżeli nie ma podanej gminy, to oznaczało, że podana miejscowość stanowi także gminę. Na potrzeby późniejszego połączenia wyciągniętych gmin z tabelą z kodami gmin z danych z GUSu usuwamy również niepotrzebne spacje oraz znaki zapytania. Sortujemy również wczytane dane, aby później dokonać merga z tabelą zawierającą dodatkowe informacje o gminach.

Wczytujemy również plik csv zawierający informację o mapowaniu gmin, a błędne rekordy przekierowujemy do error loga. Uzyskane dane również sortujemy.

Dokonujemy operacji merge dwóch wcześniej omówionych tabel. Jest to left join, ponieważ to gminy chcemy przydzielić do tabeli z masztami. Po dokonaniu złączenia poprawiamy jeszcze kolumnę z miejscowością, tak aby nie zawierała informacji o gminie, ponieważ ta jest już w oddzielnej kolumnie. Następnie zapisujemy dane do bazy danych do tabeli Maszty, a błędne rekordy kierujemy do error log.

3.3.2 Załadowanie danych do tabeli faktów GminaInfo

Dane potrzebne do załadowania do tabeli GminaInfo mamy na różnym poziomie szczegółowości, dlatego też proces ten ma za zadanie je połączyć. Wszystkie dane pochodzą z plików płaskich CSV. Tabele zawierające informacje o gęstości zaludnienia, płci i strukturze wiekowej są udostępniane na poziomie poszczególnych gmin. Te trzy tabele wczytujemy w procesie ETL, a błędne rekordy przekierowujemy do error log. Następnie sortujemy zarówno tabelę gęstość zaludnienia jak i tabele ze strukturą wiekową oraz z płciową. Po posortowaniu dokonujemy złączenia tych trzech tabel za pomocą funkcji merge (najpierw łączymy gęstość zaludnienia ze strukturą płciową, a następnie tak powstałą tabelę łączymy jeszcze ze strukturą wiekową).

Dane o bezrobociu są udostępniane na poziomie powiatów. Wczytujemy te dane, sortujemy i ponownie dokonujemy złączenia utworzonej wcześniej tabeli z tabelą zawierającą informacje o bezrobociu.

Kolejnym krokiem jest wczytanie tabeli z informacjami o mapowaniu gmin. Tak jak poprzednio sortujemy tę tabelę, aby merge był możliwy. Po posortowaniu dokonujemy złączenia z wcześniejszą tabelą. Tak powstałą tabelę przygotowujemy do kolejnego złączenia, czyli sortujemy ją.

Dane o wyposażeniu ludności w telefony i smartfony udostępniane są na poziomie województwa. Wczytujemy zatem tę tabelę ją sortujemy. Następnie dokonujemy złączenia z wcześniej utworzoną tabelą. Tak powstałą tabelę sortujemy, aby była przygotowana do kolejnego merge.

Ostatnimi informacjami jakie chcemy dołączyć jest poziom wykształcenia ludności. Niestety dane te są zbierane przez Gus tylko w Narodowych Spisach Powszechnych, przez co są to dane z 2011 roku. Udostępniane są one na poziomie powiatów. Wczytujemy zatem plik zawierający informację o wykształceniu. Następnie dokonujemy wyliczenia kolumn z id powiatu i województwa, a w przypadku niepowiedzenia przekierowujemy rekordy do error log. Później sortujemy tabelę i dokonujemy złączenia z wcześniej utworzoną tabelą.

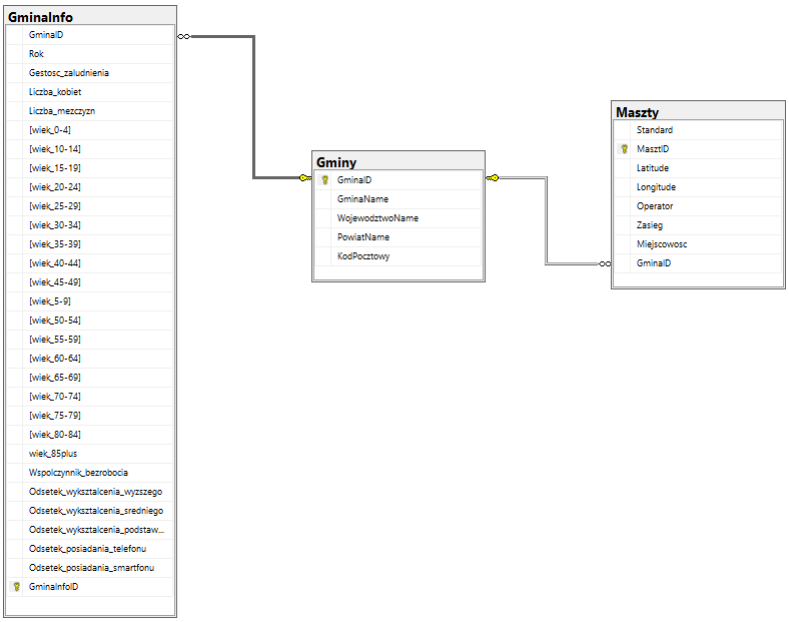
Otrzymujemy zatem jedną dużą tabelę zawierającą szereg informacji. Wyliczamy jeszcze na niej procentowe informacje o wykształceniu, a także współczynnik bezrobocia. Dane z Gusu zawierały liczbę osób na gminę, a nie informację o odsetku. Następnie sortujemy i wyliczamy gmina id. Na koniec zapisujemy rekordy do bazy danych do tabeli GminaInfo, a błędne rekordy kierujemy do error log.

3.3.3 Załadowanie danych do tabeli wymiarów Gminy

Dane wczytywane są z pliku płaskiego CSV. W przypadku wystąpienia błędnych rekordów, kierowane są one do error log. Następnie za pomocą wyliczanej kolumny ustawiamy kodowanie kolumn ze stringami na UTF-8 (z jakiegoś powodu ETL nie przechodził bez tego, pomimo ustawianie kodowania już na etapie wczytywania pliku). Później kasujemy wszystkie polskie znaki. Tak przetworzone dane zapisujemy do bazy danych do tabeli Gminy, a w przypadku błędów, przekierowujemy rekordy do error log.

3.4 Model hurtowni danych

Na rysunku 2 przedstawiamy model hurtowni danych. Hurtownia ta jest zbudowana w modelu gwiazdy. Tabelami faktów w naszej bazie danych są Maszty i GminaInfo. Ponadto znajduje się w niej jedna tabela wymiarowa - Gmina. Tabela Maszty zawiera informacje o masztach telefonicznych takie jak ich współrzędne geograficzne, standard, miasto, zasięg, czy nazwa operatora. Z kolei tabela GminaInfo zawiera podstawowe informacje o mieszkańcach danej gminy. Są to między innymi liczba kobiet i mężczyzn, gęstość zaludnienia, współczynnik bezrobocia, informacje o wykształceniu, czy informacje o posiadaniu telefonów przez mieszkańców.



Rysunek 2: Model hurtowni danych

3.5 Business Intelligence

W naszym projekcie wykorzystaliśmy Tableau jako narzędzie Business Intelligence. Z poziomu Tableau łączymy się z hurtownią danych w SQL Server. Po połączeniu przygotowaliśmy kilkanaście kluczowych z naszego punktu widzenia wizualizacji. W celu zwizualizowania danych na mapie

zamieniliśmy typy danych na geography. W szczególności, w Tableau można na mapie zaznaczać województwa mając ich nazwy oraz zaznaczać jednostki terytorialne na podstawie kodu pocztowego. Dodatkowo, utworzyliśmy wyliczane pole typu geograficznego, które odpowiadało zasięgowi danemu masztu na podstawie jego standardu. Inne utworzone przez nas wyliczane pola, to liczba osób na jeden masz w jednostce terytorialnej oraz powierzchnia jednostki terytorialnej (ponieważ mieliśmy jedynie liczby ludności i gęstości zaludnienia w gminach). Utworzyliśmy hierarchię zawierającą województwa, powiaty i gminy. Stworzyliśmy także specjalne filtry poprzez kreator parametrów i pól wyliczanych, filtry te dotyczyły wieku ludności oraz wykształcenia. W rezultacie tworzymy szereg wykresów i wizualizacji pozwalających na przeanalizowanie możliwych nowych lokalizacji masztów BTS.

3.6 Wnioski biznesowe

Na podstawie przygotowanych raportów można wyciągnąć kilka wniosków biznesowych:

- być może warto zainwestować w budowę nowych masztów w województwie Podkarpackim z uwagi na dużą liczbę osób na jeden maszt oraz wysoki odsetek posiadania smartfonu;
- warto również przeanalizować obszary o dużej gęstości zaludnienia i małej liczbie masztów takie jak: Piastów, Ząbki, Legionowo, Świętochłowice, Mińsk Mazowiecki jako potencjalne lokalizacje nowych masztów;
- wskaźnik bezrobocia nie koreluje z liczbą masztów (telefon jest już a tyle tani, że każdy może go mieć);
- wpływ wykształcenia należałoby dokładniej przeanalizować za pomocą modeli statystycznych.

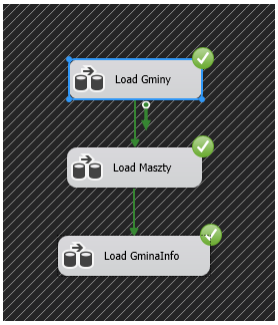
Należy jednak zaznaczyć, że przygotowane przez nas wnioski powinny zostać zweryfikowane przez analityków biznesowych. Przed podjęciem decyzji o budowie nowego masztu powinno się przeprowadzić bardziej dogłębną analizę skupiającą się na konkretnym obszarze.

3.7 Testy funkcjonalne

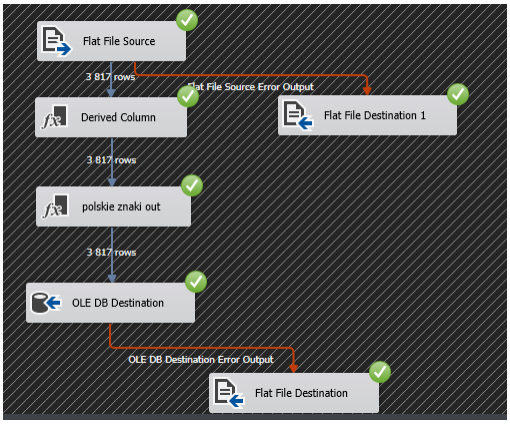
W tej sekcji omówimy przeprowadzone przez nas testy funkcjonalne.

3.7.1 Wczytywanie danych do pustej tabeli

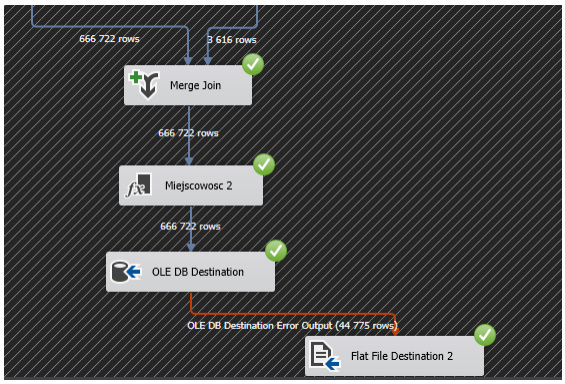
Pierwszy test ma za zadanie zweryfikować, czy zasilenie hurtowni danych działa poprawnie. Rozważamy tutaj sytuację, gdy mamy początkowo pustą hurtownie danych i dostajemy dane do jej zasilenia. Zauważmy, że test ten zakończył się sukcesem, co widzimy na rysunkach 3, 4, 5 i 6.



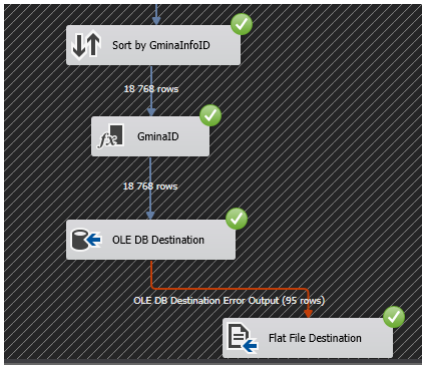
Rysunek 3: Wczytanie do pustej tabeli - control flow



Rysunek 4: Wczytanie do pustej tabeli - tabela Gminy



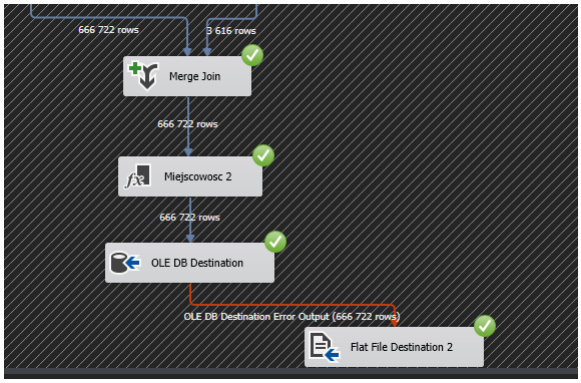
Rysunek 5: Wczytanie do pustej tabeli - tabela Maszty



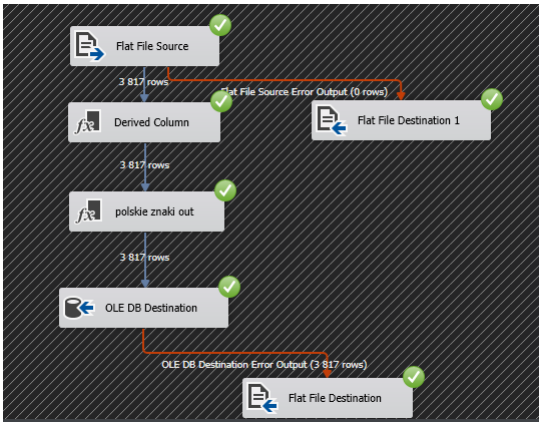
Rysunek 6: Wczytanie do pustej tabeli - tabela GminaInfo

3.7.2 Wczytywanie dokładnie takich samych rekordów do tabeli

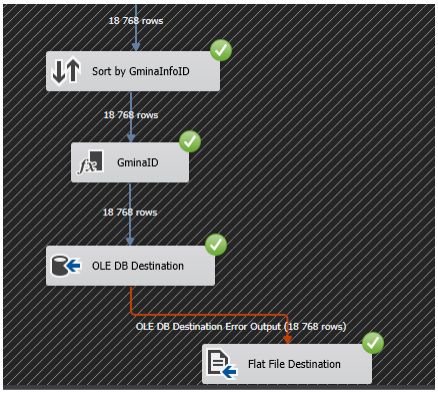
W tym teście sprawdzamy, czy scenariusz, że hurtownia danych zawiera już jakieś rekordy i próbujemy do niej wczytać dokładnie te same rekordy. Zauważmy, że test ten zakończył się sukcesem, co widzimy na rysunkach 7, 8 i 9.



Rysunek 7: Wczytanie istniejących wierszy - tabela Maszty



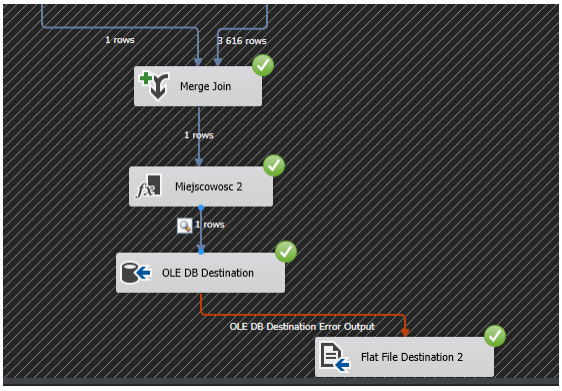
Rysunek 8: Wczytanie istniejących wierszy - tabela Gminy



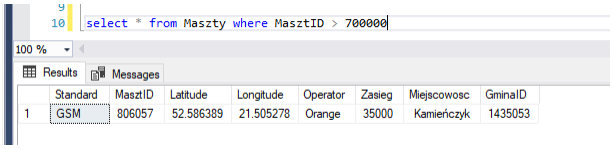
Rysunek 9: Wczytanie istniejących wierszy - tabela GminaInfo

3.7.3 Wczytywanie nowego rekordu

Ten test ma za zadanie sprawdzić, czy można wczytać nowy rekord do hurtowni danych. W sytuacji początkowej hurtownia zawiera już jakieś rekordy. W teście tym chcemy wstawić nowy rekord. Zauważmy, że test ten zakończył się sukcesem, co widzimy na rysunkach 10 i 11.



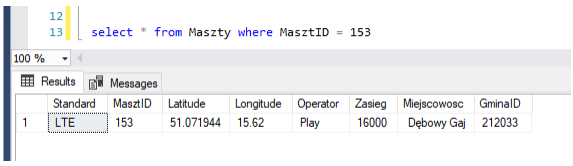
Rysunek 10: Wczytanie nowego rekordu



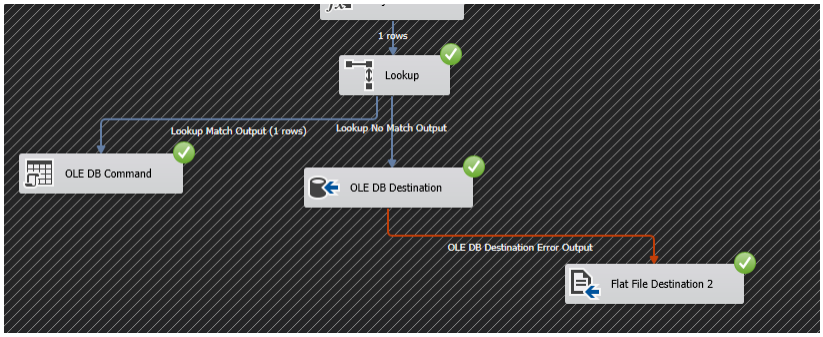
Rysunek 11: Nowy rekord w bazie danych

3.7.4 Wczytywanie rekordu zawierającego inne dane, ale o tym samym kluczu głównym

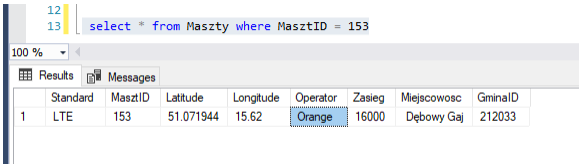
Test ten ma za zadanie sprawdzić, czy w sytuacji gdy mamy już rekord zawierający dane w hurtowni, możemy go uaktualnić. Zauważmy, że test ten zakończył się sukcesem, co widzimy na rysunkach 12, 13 i 14



Rysunek 12: Rekord przed uaktualnieniem



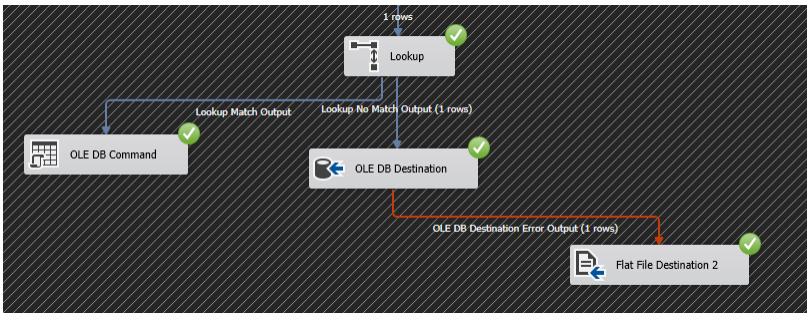
Rysunek 13: Wykonanie uaktualnienia informacji o maszcie



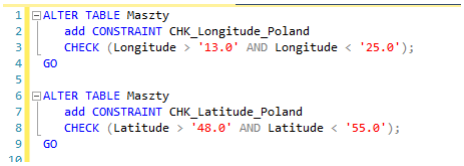
Rysunek 14: Rekord po uaktualnieniu

3.7.5 Wczytywanie rekordu zawierającego informacje o maszcie poza granicami Polski

W tym teście próbujemy wstawić rekord z informacją o nowym maszcie, znajdującym się poza granicami Polski. Sprawdzenia dokonujemy za pomocą sprawdzenia granicznych wartości szerokości oraz długości geograficznej. Test ten zakończył się sukcesem, co widzimy na rysunkach 15 i 16



Rysunek 15: Wczytanie masztu znajdującego się poza granicami Polski



Rysunek 16: Wczytanie masztu znajdującego się poza granicami Polski - ograniczenia

3.8 Podsumowanie

Całość prac nad projektem nauczyła nas kilku rzeczy. Po pierwsze poznaliśmy ciekawe rozwiązanie klasy Business Intelligence - Tableau. Stwierdzamy, że jest to bardzo intuicyjne i przydatne rozwiązanie do tworzenia raportów biznesowych. Pozwala ono na stworzenie ciekawych wizualizacji i dashboardów. Ponadto można stwierdzić, że pomimo kilku trudności proces ETL przebiegł pomyślnie i pozwolił zasilić naszą bazę danych w poprawny sposób. Warto jednak zauważyć, że SSIS jest to bardzo kompleksowe rozwiązanie i wymaga bardziej pogłębionej nauki, aby móc zrozumieć wszystkie jego tajniki (poznanie przyczyny niedziałania funkcji pivot).