Hurtownie danych i systemy Business Intelligence 2021

ANALIZA RYNKU PRACY W CZASIE TRWANIA PANDEMII COVID-19

7 czerwca 2021

Ada Gąssowska, Mikołaj Jakubowski, Elżbieta Jowik

Contents

1 Opis projektu

1.1 Założenia projektu

Generalnym zadaniem hurtowni danych jest uporządkowanie tematyczne krytycznych, z punktu widzenia organizacji, obszarów analitycznych, ujednolicenie informacji oraz udostępnienie ich do analizy zagadnień decyzyjnych. Przygotowany projekt nosi znamiona uproszczonego modelu i ogranicza analizę do lokalizacji tylko w obrębie Stanów Zjednoczonych agregując przy tym dane pochodzące z trzech różnych źródeł:

- dwóch różnych API przechowujących historyczne i bieżące informacje nt. publikowanych ofert pracy (*Careerjet API*, *Jooble API*),
- repozytorium *The New York Times*, na którym dostępne są codziennie aktualizowane statystyki dot. liczb zakażeń i zgonów dla poszczególnych hrabstw w Stanach Zjednoczonych.

Celem projektu było przeprowadzenie analizy danych o ofertach pracy, dostępnych w Stanach Zjednoczonych, z uwzględnieniem badania ich zmienności pod wpływem rozwoju pandemii SARS-CoV-2 a także zindywidualizowane badania każdego z wymienionych obszarów, tj. amerykańskiego rynku pracy i dynamiki przebiegu pandemii COVID-19 w tym rejonie świata.
Produktem końcowym miała być hurtownia danych umożliwiająca analizę zróżnicowania wpływu pandemii (ze względu na miejsce) na ilość i ogólnie pojmowaną jakość dostępnych ofert oraz przeprowadzenie badania dynamiki rynku w określonej lokacji.

1.2 Wartość biznesowa

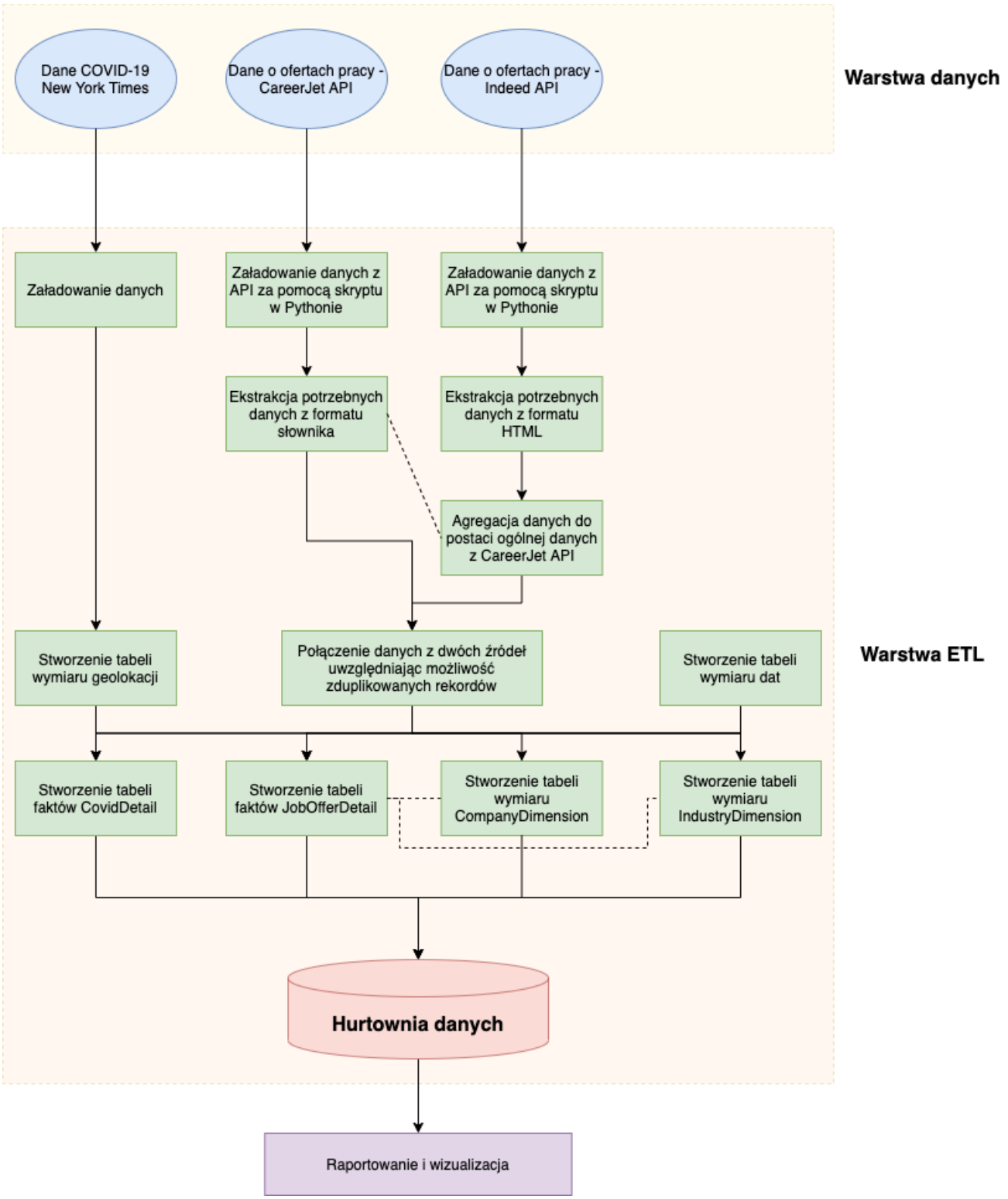
Przygotowana hurtownia danych pozwala na pozyskanie wielu istotnych informacji o rynku pracy, w tym odpowie na pytania:

- Jakie sektory zatrudnienia są najsilniejsze w danym miejscu/czasie?
- Jakie lokalizacje warto rozważyć przy wyjeździe zarobkowym?
- Czy na bieżącym etapie przebiegu pandemii skala zachorowań w danym miejscu wpływa na dynamikę rynku pracy?

Taka hurtownia pozwala wygenerować istotne biznesowo informacje, które mogą być interesujące z punktu widzenia przedstawicieli portali rekrutacyjnych czy agencji pracy. Dodatkowo duża aktywność na rynku pracy w danej lokalizacji może świadczyć o rozwojowości lokacji i jej potencjale inwestycyjnym, a z kolei ta informacja może okazać się przydatna dla firm poszukujących biznesowego uzasadnienia lokowania przedsiębiorstw.

2 Architektura rozwiązania

2.1 Diagram



Rysunek 1: Diagram architektury

2.2 Opis architektury

Strukturę projektowanej hurtowni danych tworzą kolejne warstwy, spośród których każda następna warstwa jest bezpośrednim przetworzeniem poprzedniej.

Pierwszą warstwę tworzą źródła danych, czyli zastane dane, pochodzące z trzech niezależnych źródeł. Są one zróżnicowane pod względem sposobu dostępu (ogólnodostępny plik .csv, dane pozyskiwane za pośrednictwem interfejsów API), struktury logicznej, wielkości i jakości.

Centralną warstwą na schemacie jest warstwa ETL, obejmująca: ekstrakcję danych ze źródeł, ich transformację do pożądanej postaci, integrację i załadowanie do hurtowni. W pierwszej fazie surowe dane są pobierane z rozproszonego systemu źródłowego i wyodrębniane są z nich informacje krytyczne z punktu widzenia projektu - ekstrakcja ta dotyczy w szczególności danych pozyskiwanych za pomocą dedykowanych interfejsów. W następnym etapie, dane pozyskane za pośrednictwem *Jooble API*, zostają zagregowane do postaci słownikowej celem zachowania spójności danymi z *CareerJet API* i systemem docelowym. Po przeprowadzeniu transformacji następuje konwersja wszystkich danych na ujednolicone formaty i typy, integracja danych dotyczących ofert pracy i agregacja do struktury narzuconej przez docelowy model hurtowni i kontekst analizy. Ostatnim etapem procesu ETL jest ładowanie danych do docelowego repozytorium danych, którym w bieżącym projekcie jest zaproponowana hurtownia.

Najniższa warstwa tj. *Raportowanie i wizualizacja* jest warstwą eksploatacyjną. Udostępnia ona końcowemu użytkownikowi biznesowemu narzędzia raportowo analityczne i usługi dostępu do danych hurtownianych.

3 Zbiory danych

Dane do hurtowni pozyskiwane są z trzech, wymienionych we wstępie, źródeł. Pierwsze dwa dotyczą danych o ofertach pracy, trzecie źródło dotyczy danych o pandemii COVID-19.

Careerjet to wyszukiwarka pracy, która pobiera oferty pracy ze stron internetowych firm i organizacji w różnych branżach i lokalizacjach. Wyniki wyszukiwania mogą być zwracane według słów kluczowych, lokalizacji, firm, branż i innych kryteriów. Odpowiedzią na żądanie wysłane do *Careerjet API* jest JSON, w którym każda oferta pracy jest zdefiniowana przez: nazwy stanowiska i firmy, datę publikacji, lokalizację, link będący bezpośrednim przekierowaniem do oferty, opis stanowiska, wysokość i walutę pensji. Spośród wymienionych pól, z konieczności ujednolicenia struktur danych pochodzących z dwóch interfejsów programowania aplikacji, w tabeli faktowej nie zostały uwzględnione: opis i waluta, przy czym charakterystyka stanowiska została wykorzystana w procesie transformacji danych do przypisania branż wybranym ofertom.

Jooble API zwraca listę wszystkich, aktywnych w momencie wysłania żądania, ofert pracy dostępnych na tablicy ogłoszeń. Ich filtrowanie również jest możliwe przy użyciu opcjonalnych parametrów ciągu zapytania a zwracana lista jest posortowana według daty publikacji na tablicy. Dane pozyskane za pośrednictwem interfejsu *Jooble API*: są typu *bytes* a poszczególne oferty są parametryzowane przez: id, nazwę i opis stanowiska, lokalizację, wysokość pensji, źródło, wymiar etatu, źródło, z którego oferta została pobrana, link będący bezpośrednim przekierowaniem do oferty oraz datę publikacji. O umieszczeniu danego pola w tabeli faktowej decydowała zgodność jego charakteru z polami wybranymi z danych *Carreerjet*.

W powyższych danych nie jest określona jednolita konwencja dla reprezentacji wysokości wynagrodzenia - możliwe postaci obejmują różne formy agregacji np. minimum, przedział lub wartość podana wprost. Problem ten jest obsługiwany w fazie transformacji w procesie ETL.

Dane na temat pandemii COVID-19 w Stanach Zjednoczonych, zbierane przez *The New York Times*, mają bardzo prosty i łatwy do odczytu format. Ramka składa się z kolumn: daty, hrabstwa, stanu, liczby zachorowań oraz liczby zgonów.

4 Faza ETL

4.1 Pozyskanie danych ze źródeł zewnętrznych

Czynnością bezpośrednio poprzedzającą proces ekstrakcji danych ofert pracy było przygotowanie danych, zawierających spis 314 amerykańskich miast o największej populacji. Zostały one wygenerowane w oparciu o informacje opublikowane na Wikipedii, za pomocą dostępnego online narzędzia konwertującego tabele dostępne na tej stronie tabele do plików .csv. Tak wygenerowane dane zostały wczytane do skryptu Pythona i jako ramka danych wykorzystane w procesie zczytywania ofert z każdego z interfejsów programowania aplikacji. Dzięki temu, przy użyciu, umieszczonego w ciągu zapytania, opcjonalnego parametru lokalizacji, możliwe było filtrowanie dostępnych danych już na etapie ich ekstrakcji. W konsekwencji w hurtowni danych umieszczone zostały, tylko te oferty, których wartości pola lokalizacji były zgodne z założeniami projektu, tj. obejmowały tylko miasta Stanów Zjednoczonych. Dane dostępne za pośrednictwem API są zagregowane w strony, w związku z czym ich zczytywanie i jednoczesna selekcja rekordów zostały wykonane w podwójnej pętli (iterując po kolejnych lokalizacjach i wszystkich dostępnych dla nich stronach) również z poziomu skryptu, napisanego w języku Python.

Pozyskanie danych z API serwisu *Jobble* wymagało rejestracji w serwisie i pozyskania klucza autoryzującego, umożliwiającego dostęp do danych.

Dane dotyczące liczb zakażeń i zgonów dla poszczególnych hrabstw w Stanach Zjednoczonych zostały pobrane w formie pliku .csv bezpośrednio z repozytorium *The New York Times*, na którym dostępne są codziennie aktualizowane statystyki.

4.2 Opis niezbędnych transformacji danych w procesach ETL

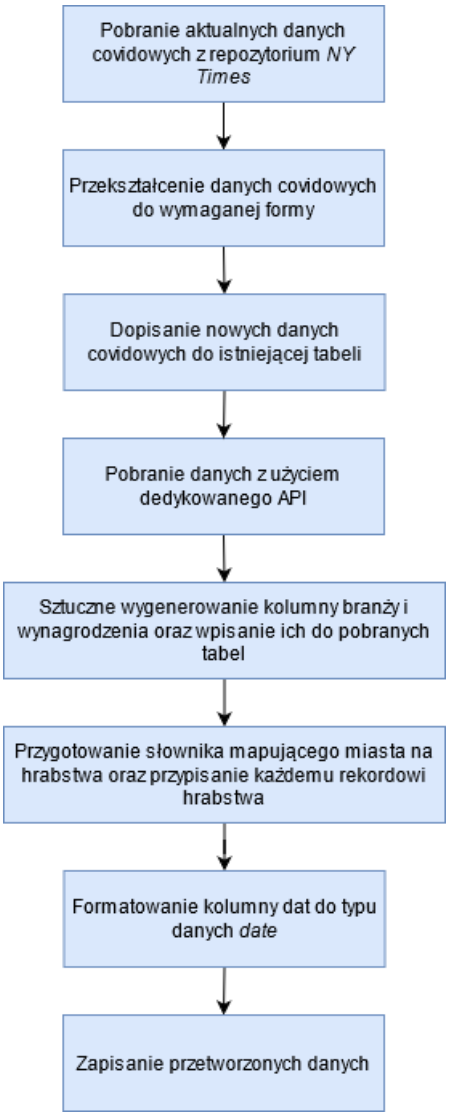
Dane pobrane z API Careerjet mają postać zagnieżdzonego słownika (format JSON), w którym każda oferta pracy jest obiektem klasy dict, natomiast interfejs Jooble API zwraca wygenerowane dane w postaci zmiennej bajtów (typ *bytes*), które na potrzeby integracji są przetwarzanie z poziomu skryptu w Pythonie.

Dane zawierające informacje o zachorowaniach w pandemii również są pobierane z użyciem tego skryptu, z publicznego repozytorium *New York Times*. Liczba przypadków oraz śmierci zapisane w tej tabeli, są jednak zapisane w postaci zkumulowanych sum do każdego dnia. Jako jeden z kroków w procesie transformacji, dane te są przekształcane do postaci niezkumulowanej.

Na potrzeby projektu wspomniany skrypt w Pythonie zajmuje się również sztucznym uzupełnianiem trudno dostępnych danych. Kolumny takie jak zarobki i branża dla ofert pracy, są generowane losowo z odpowiednich rozkładów. Dodatkowo oferty dla branży 'IT', wyszukiwane są za pomocą ustalonej bazy słów kluczy z nią związanych. Dokładna ekstrakcja takich informacji wprost z pobranych tabel, okazuje się wymagać zastosowania specjalistycznych algorytmów, które wykraczają poza cele tego projektu. Należy zatem założyć, że opisywana struktura jest jedynie prototypem, którego działanie przedstawiamy na częściowo sztucznych danych, który przy dostępności danych prawdziwych poradziłby sobie równie dobrze.

Zakładamy, że w przypadku braku danych, w którymkolwiek z pól kluczowych, z punktu widzenia projektowanej hurtowni (w szczególności data publikacji, lokalizacja), dana oferta pracy będzie eliminowana w fazie ładowania. Jeśli natomiast braki danych będą występowały dla pól mniej krytycznych, to dla danej oferty będą one wypełniane dedykowanymi brakom wartościami, np. 0 lub *unknown*.

Dla lepszego zobrazowania kolejnych kroków podejmowanych w przetwarzaniu danych w skrypcie, poniżej znajduje się diagram prezentujący owy proces. W celu dogłębnego poznania działania skryptu, zaleca się bezpośrednie zaopznanie z kodem, zawierającym dodatkowe komentarze do każdego z kroków.



Rysunek 2: Diagram przedstawiający kolejne kroki skryptu w Pythonie

W kontekście wymiaru lokalizacji, jak wspomniano we wstępie, przygotowany projekt nosi znamiona uproszczonego modelu. Za minimalną granulację danych geolokalizacyjnych przyjmuje się poziom hrabstwa. Informacje na ich temat zostały umieszczone w tabelach faktów, dzięki wykonananemu w skrypcie Python'owym (za pomocą pakietu *geocoder*) mapowaniu miast i stanów na stany i hrabstwa. W mapowaniu tym wykorzystano *API Google Maps* i ponownie przygotowaną wcześniej ramkę zawierającą dane największych, według liczby ludności, miast Stanów Zjednoczonych. Podobnie jak w przypadku *Jobble API*, połączenie z *API Google Maps* wymaga rejestracji do serwisu i uzyskania klucza autoryzacyjnego dla projektu.

Sam wymiar lokalizacji (*GeographyDim*) został stworzony na podstawie pliku csv udostępnionego i aktualizowanego codziennie przez magazyn *The New York Times*. Faza ETL dla tych danych była znacznie mniej skomplikowana - wymiar lokalizacji został bowiem wygenerowany na drodze wyboru unikatowych par hrabstwo-stan i nadania każdej takiej parze unikalnego klucza.

4.3 Ładowanie danych do hurtowni

W kroku opisanym w bieżącej sekcji niniejszej dokumentacji, następuje załadowanie tych danych, które spełniają przyjęte założenia do centralnego repozytorium, czyli docelowej hurtowni danych. W celu zachowania spójności referencyjnej, w procesie wczytywania danych zastosowano predefiniowany porządek zgodnie, z którym w pierwszej kolejności zasilono danymi tabele wymiarów a

następnie tabele faktów.

Ze względu na brak zmienności i tym samym brak konieczności aktualizacji danych przechowywanych w tabelach wymiarów daty i geografii, tabele te zostały zasilone jednorazowo za pośrednictwem zintegrowanego środowiska *Microsoft SQL Management Studio*.

Pozostałe tabele zasilono z poziomu *Integration Servics Project* utworzonego za pomocą narzędzia *Microsoft Visual Studio*. Podczas wczytywania danych z plików źródłowych zastosowano podejście polegające na odrzuceniu rekordu i jego przekierowywaniu do zewnętrznego pliku tekstowego w przypadku przycięcia wartości któregokolwiek z pól.

Ponadto, biorąc pod uwagę systematyczność aktualizacji danych w hurtowni, przed załadowaniem danego rekordu, do którejkolwiek tabeli sprawdza się, czy nie został on już do niej wcześniej załadowany. Na przykład dla danych covidowych sprawdza się czy w przechowującej je tabeli istnieje już rekord o lokalizacji i dacie jednakowej jak w rekordzie wczytywanym - jeśli tak, wówczas taki rekord jest odrzucany. Dzięki temu minimalizuje się ryzyko powielenia lub każdorazowego nadpisywania jakichkolwiek danych, w szczególności danych historycznych.

Co więcej każdy rekord badany jest też pod kątem obecności wśród jego wartości obiektów pustych (NullObject) lub pustych pól tekstowych. W konsekwencji obserwacje z brakami danych w którymkolwiek z pól kluczowych, z punktu widzenia projektowanej hurtowni, które zgodnie z przyjętą metodologią nie są wypełniane dedykowanymi brakom wartościami, będą eliminowane w fazie ładowania.

Warto zaznaczyć, że biorąc pod uwagę ryzyko istnienia tych samych ofert w obydwu źródłach, na etapie wczytywania przetransformowanych danych następuje odrzucenie duplikatów ofert, gdzie za oferty identyczne uznaje się te o jednakowych nazwach stanowisk, lokalizacji i datach publikacji. W przepływie danych do tabel wymiarów dodatkowo sprawdza się, czy wśród samych danych na bieżąco wczytywanych występują rekordy zduplikowane - jeśli tak są one eliminowane w odpowiednim *Data Flow*.

W obrębie każdego *DataFlow* następuje również mapowanie wartości niektórych zmiennych na przypisane tym wartościom ID - w szczególności dotyczy to nazw branży, firmy i hrabstwa oraz pola daty publikacji. Wymiary z punktu widzenia tabel faktowych w proponowanej architekturze, w pewnym sensie pełni rolę danych referencyjnych - zbioru danych, określającego zbiór dopuszczalnych wartości, które mogą w danych występować i zostać zmapowane na ID - jeśli takie mapowanie dla danego rekordu nie jest możliwe, jest on przekierowywany do odpowiedniego pliku zewnętrznego.

5 Model hurtowni danych wraz z opisem poszczególnych komponentów

W rozważanym projekcie zastosowana zostala struktura konstelacyjna (konstelacja faktów), bazująca na strukturze gwiazdy, w której obecne są dwie tabele faktów a wymiary daty i lokalizacji są współdzielone. Jako tabele faktów przyjęto tabele *JobOffer Detail* i *COVID Detail* przechowujące informacje o odpowiednio zmienności rynku pracy i przebiegu pandemii w czasie. Z punktu widzenia praktycznych zastosowań są to tabele podlegające częstym aktualizacjom i szybko się powiększające. Każda z tabel charakteryzuje się jednoatrybutowym, sztucznie wygenerowanym kluczem głównym oraz kluczy obcych odwołujących się do tabel wymiarów. Wymiary te ustalają kontekst analizy i są definiowane przez atrybuty tabel: *DateDimension*, *GeographyDimension*, *IndustryDimension*, *CompanyDimension*.

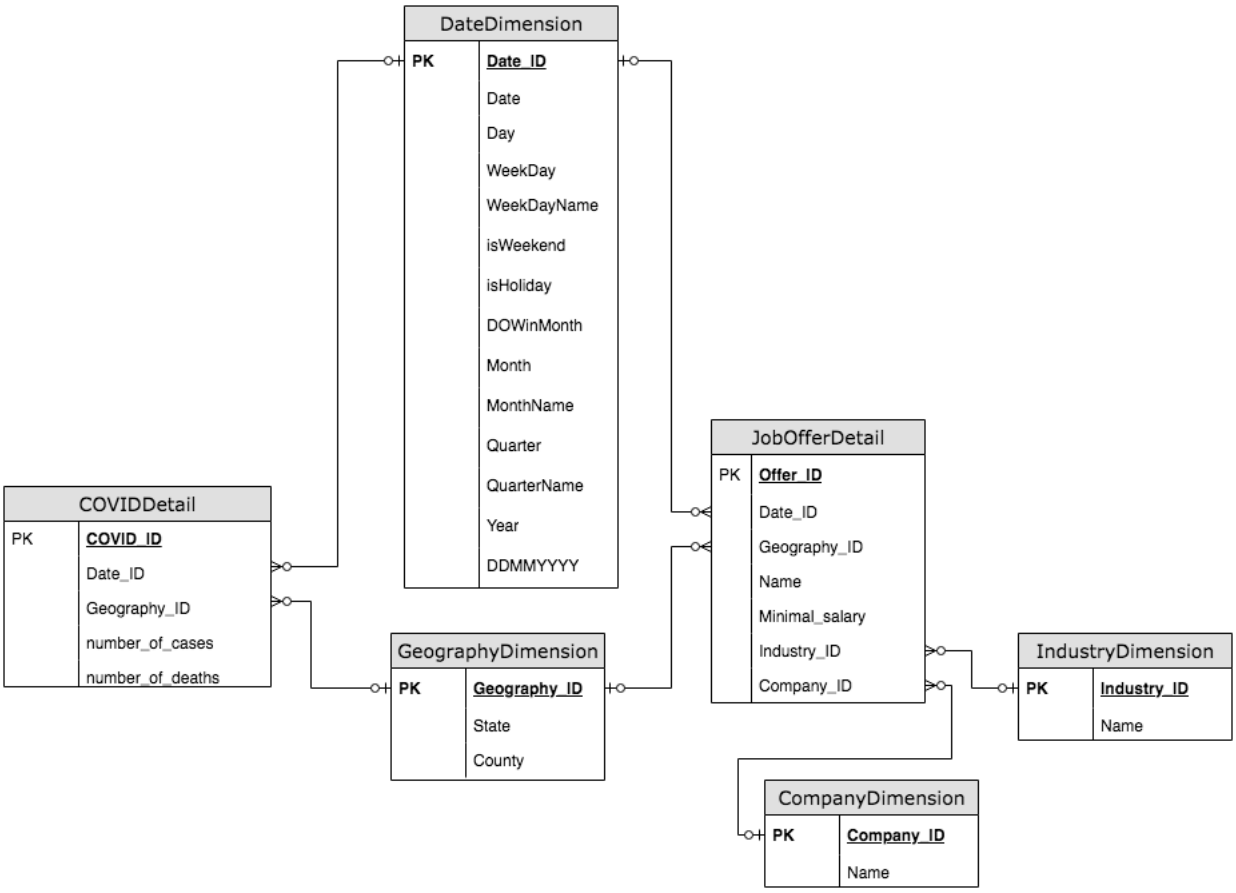
Tabela *JobOfferDetail* przechowuje dane o konkretnych ofertach prac. Oprócz odniesień do wymiarów, czyli daty wstawienia ogłoszenia, lokalizacji, branży i firmy oferującej posadę, w tabeli znajduje się nazwa ogłoszenia oraz minimalna pensja na tym stanowisku. Każde ogłoszenie posiada sztucznie generowany unikatowy klucz **OfferID**.

W tabeli faktów *CovidDetail*, znajdują się informacje dotyczące przebiegu pandemii COVID-19. Dla każdej daty i miejsca (odnośniki do tabel wymiarów), posiadamy informacje na temat liczby

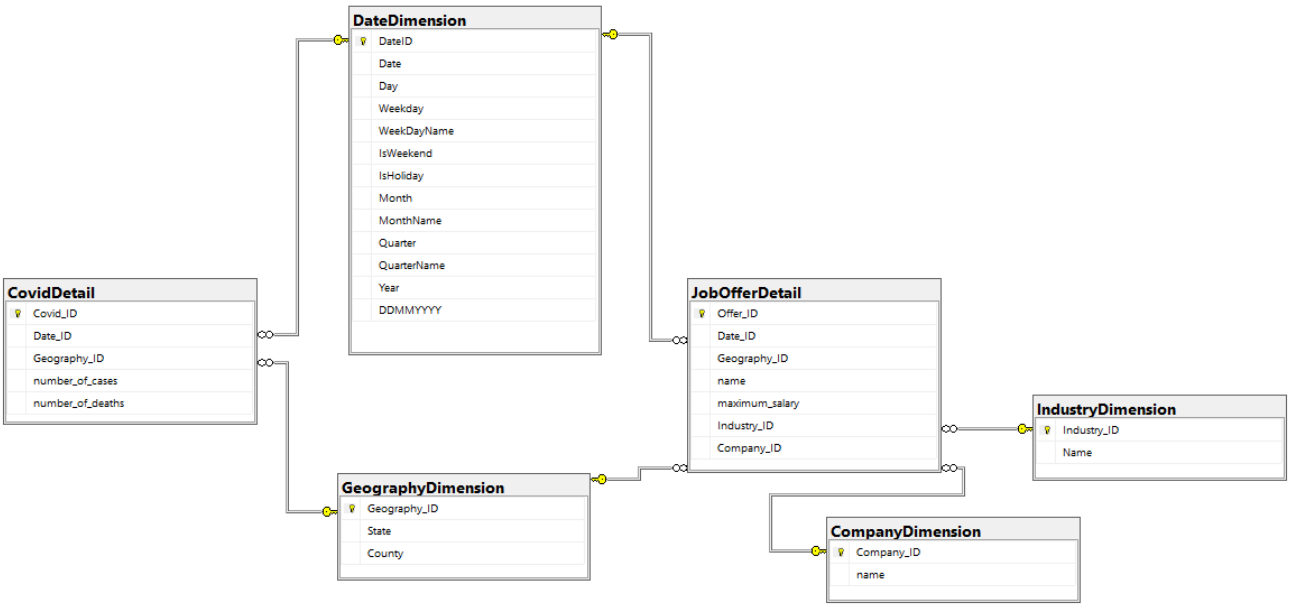
zgonów oraz liczby nowych przypadków. Każdy rekord w tabeli posiada unikatowy sztucznie generowany klucz **COVID_19**.

Dane w poszczególnych tabelach zostały ograniczone w celu zachowania prostoty struktury. Przyjęto założenie, że dodatkowe miary zagregowane będą generowane na bieżąco zgodnie z kontekstem prowadzonych analiz.

Zaproponowana struktura pozwala na stworzenie indywidualnych zestawień, w których ilość i jakość publikowanych ofert lub ilości zakażeń i zgonów mogą być prezentowane jako funkcja czasu, z możliwością ograniczenia prezentowanych danych do interesujących użytkownika okresów lub lokacji. Ponadto możliwe jest badanie wzajemnej wrażliwości rynku pracy i przebiegu pandemii COVID-19.



Rysunek 3: Diagram bazy danych przygotowany w fazie projektowania rozwiązania



Rysunek 4: Diagram rzeczywistej bazy uzyskany po wdrożeniu projektu rozwiązania

6 Opis warstwy raportowej

Końcowa warstwa raportowa, wygenerowana z wykorzystaniem narzędzia PowerBI, przedstawia dane z hurtowni oraz zależności między nimi, w sposób czytelny dla użytkownika biznesowego. Dane pobierane są za pomocą narzędzia SQL SERVER, dzięki czemu administrator ma możliwość aktualizowania danych w raporcie. Tabele faktów, tabele wymiarów oraz wszelkie połączenia między nimi są zachowane w warstwie raportowej.

Stworzone zostały dwie hierarchie - hierarchia miejsca składające się z kolejno stanów a następnie hrabstw, oraz hierarchia daty składająca się z roku, miesiąca i dnia. Dodatkowo dodana została nowa miara - 'Death Ratio' - opisująca to jakim procentem wszystkich przypadków zachorowań na COVID-19 są przypadki śmiertelne.

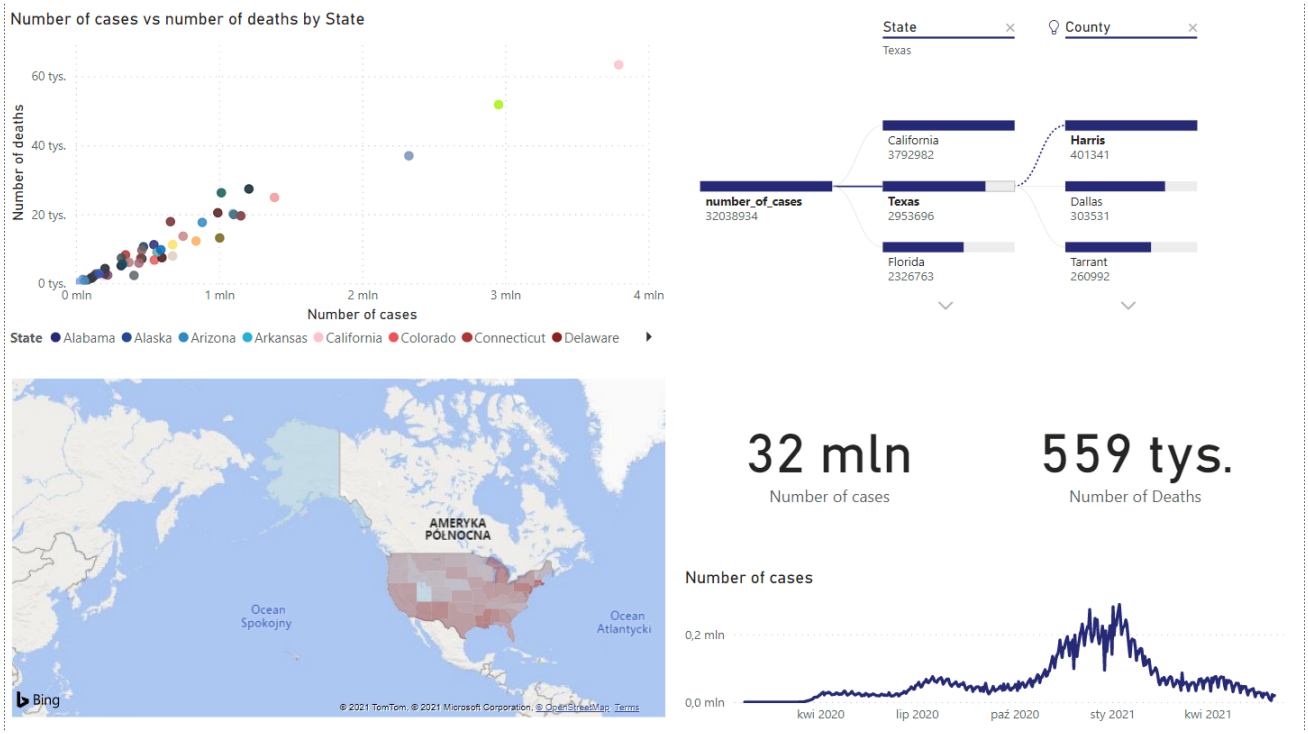
Końcowy raport składa się z trzech sekcji - sekcji odnoszącej się tylko do danych dotyczących pandemii COVID-19, sekcji odnoszącej się tylko do ofert pracy oraz sekcji dotyczącej zależności pomiędzy tymi dwoma sektorami. Warto zaznaczyć, że te trzy sekcje są realizacjami trzech, postawionych we wstępie, celów naszego projektu.

Sekcja pierwsza, dotycząca danych covidowych zaprezntowana jest na Rysunkach 5 i 6. Ukazuje ona informacje o liczbie przypadków oraz liczbie zgonów w zależności od daty i lokalizacji, które użytkownik może dowolnie filtrować przy pomocji interaktywnych wizualizacji.

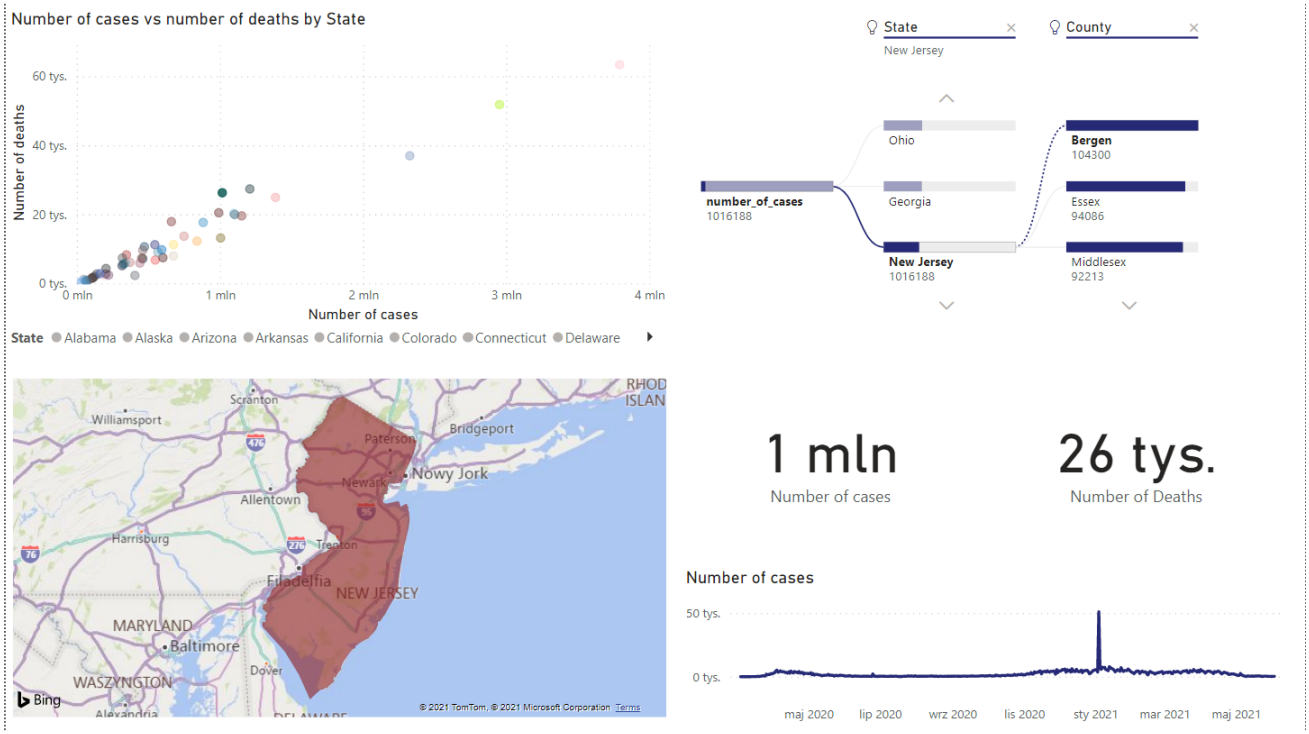
Sekcja druga, dotyczy danych o ofertach pracy i widoczna jest na Rysunkach 7 i 8. Użytkownik, za pomocą płaskich zestawień, może analizować liczbę ofert oraz średnią pensję w zależności od lokalizacji, branży oraz firmy publikującej ofertę.

Ostatnia sekcja prezentuje zależności między danymi dotyczącymi pandemii oraz danymi dotyczącymi ofert pracy. Użytkownik może filtrować dane po branży oraz lokalizacji (stanie). Wykresy przedstawiają zmienność pensji oraz ilości ofert, a użytkownik ma możliwość porównywania tych trendów z przebiegiem pandemii.

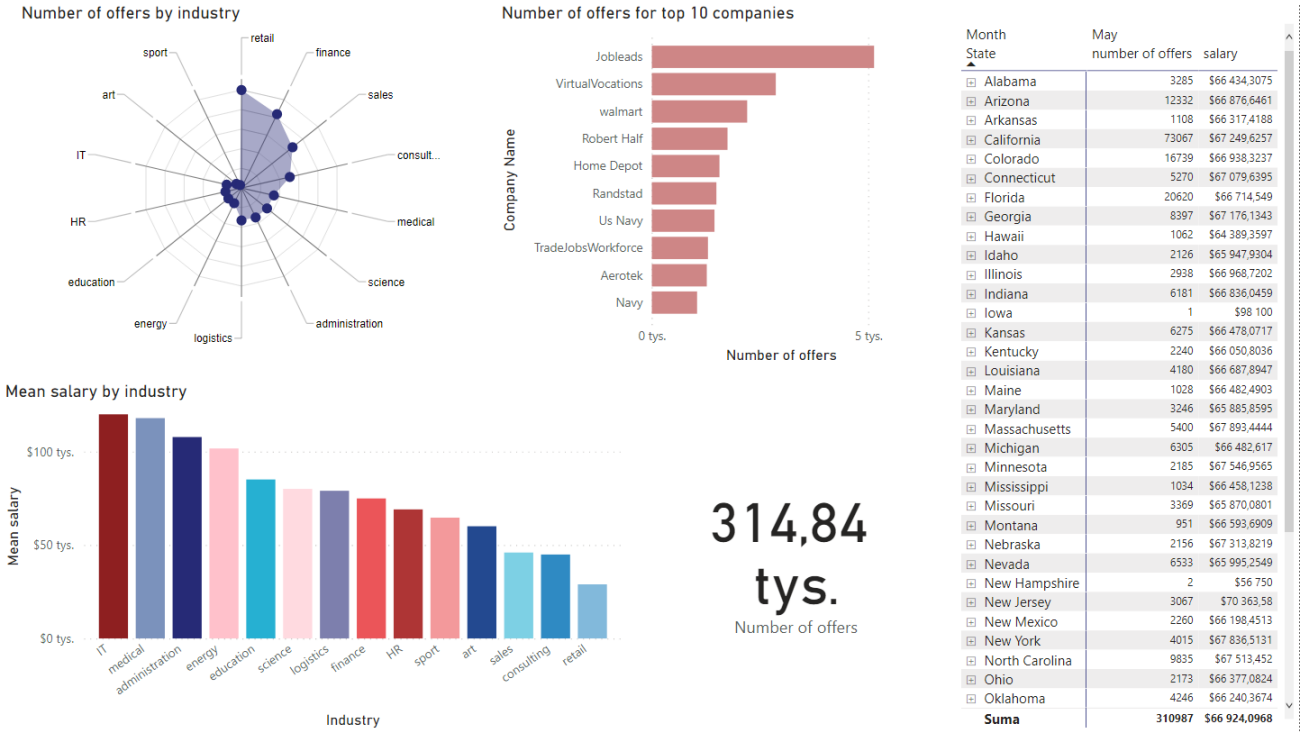
7 Prezentacja przykładowych raportów dla użytkownika



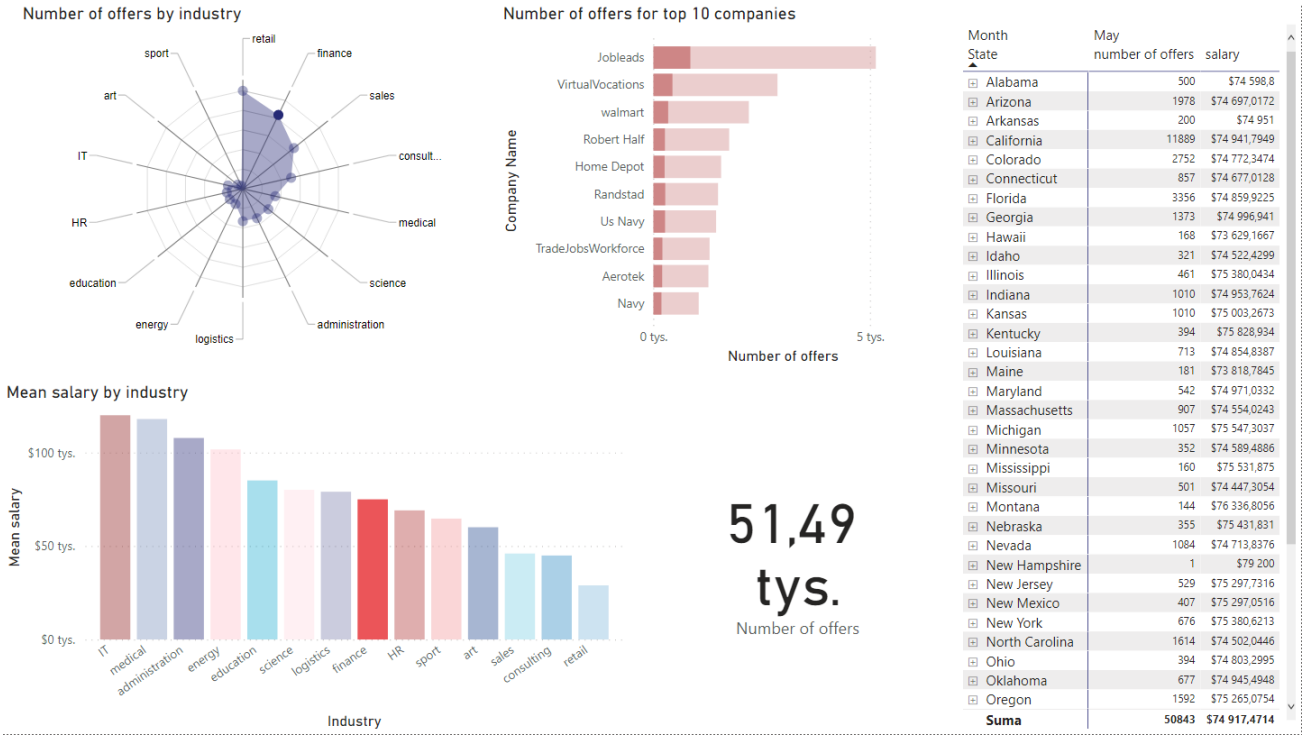
Rysunek 5: Sekcja raportu dotycząca danych covidowych



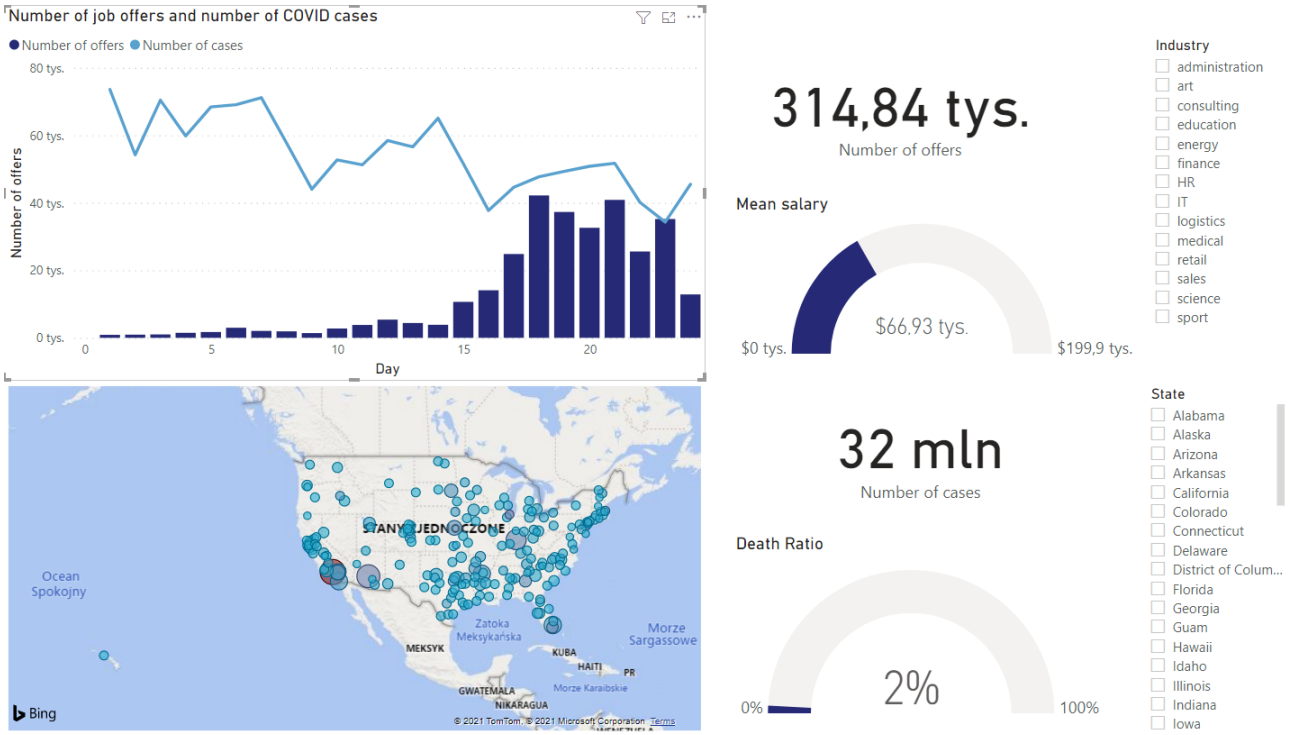
Rysunek 6: Sekcja raportu dotycząca danych covidowych, z wyróżnionym przez użytkownika stanem



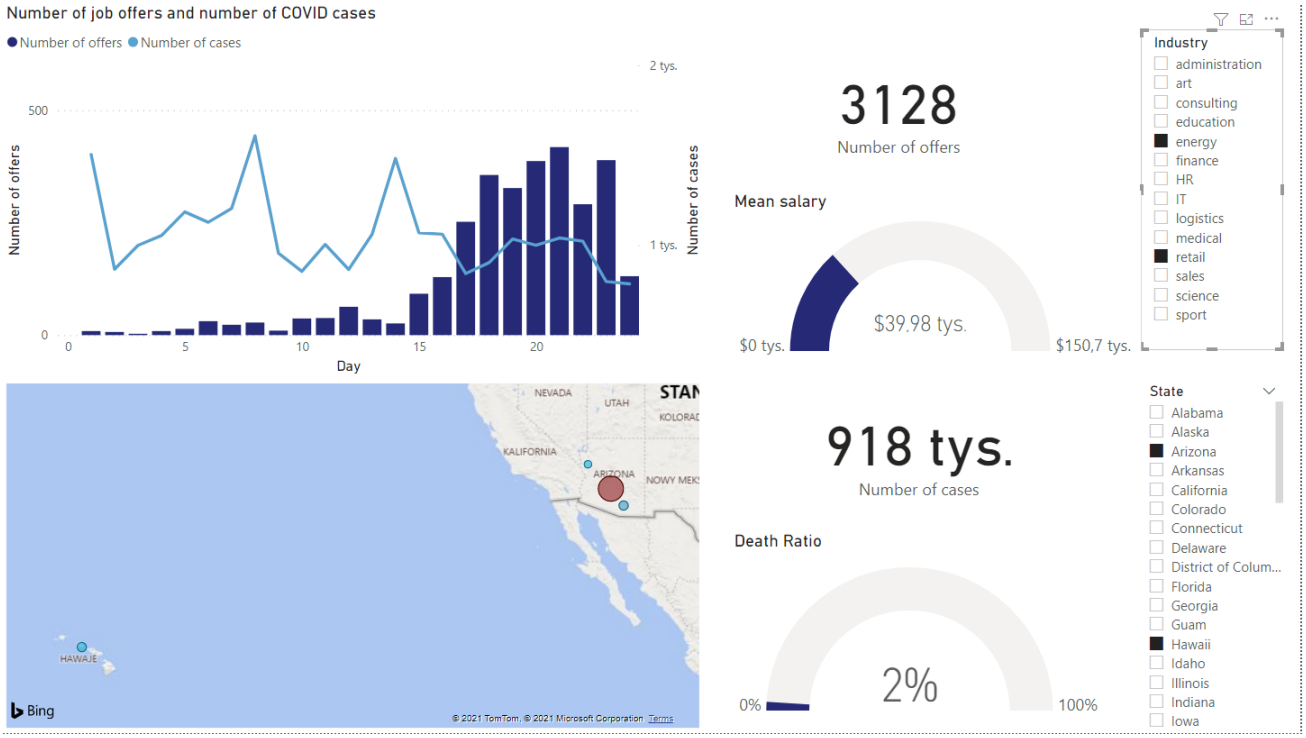
Rysunek 7: Sekcja raportu dotycząca ofert pracy



Rysunek 8: Sekcja raportu dotycząca ofert pracy z wyróżnioną wybraną przez użytkownika branżą



Rysunek 9: Sekcja raportu dotycząca zależności między ilością i jakością ofert pracy a przebiegiem pandemii



Rysunek 10: Sekcja raportu dotycząca zależności między ilością i jakością ofert pracy a przebiegem pandemii z uwzględnieniem dostępnych dla użytkownika filtrów

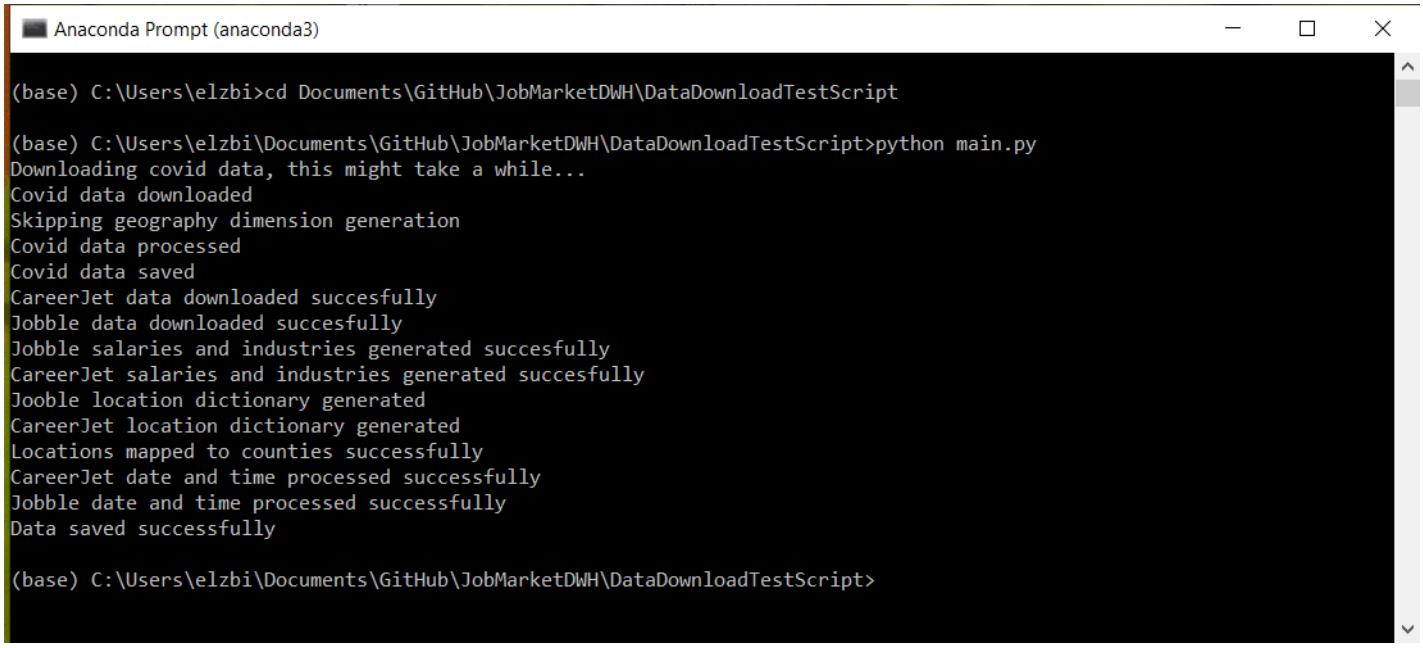
8 Podsumowanie rezultatów projektu

Zgodnie z założeniami, przygotowana hurtownia daje obszerne możliwości analizy rynku pracy w okresie pandemicznym. Umożliwia tworzenie zaawansowanych raportów biznesowych (jak ten widoczny w rozdziale powyżej). Takie informacje i różnego rodzaju zależności, mogą okazać się bardzo wartościowe z punktu widzenia portali rekrutacyjnych, agencji pracy lub nawet inwestorów poszukujących miejsca na otwarcie nowego przedsiębiorstwa.

9 Podsumowanie przeprowadzonych testów funkcjonalnych

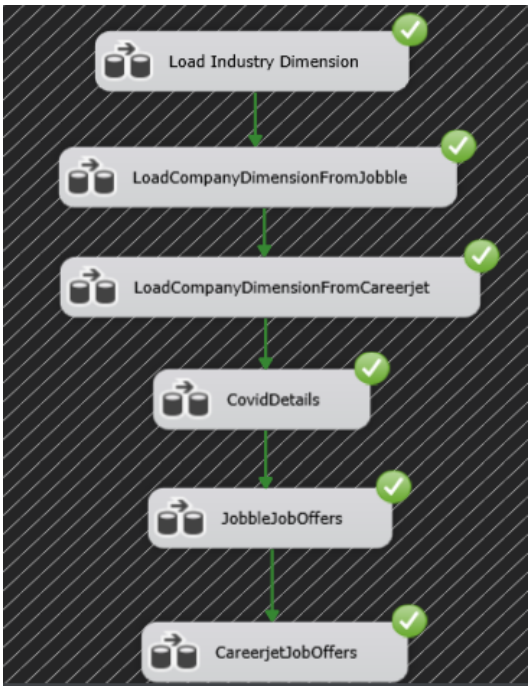
W celu weryfikacji poprawności działania różnych komponentów architektury, dla każdego z nich zostały przeprowadzone odpowiednie jednostkowe testy funkcjonalne, zarówno dla pierwszego załadowania danych jak i kolejnych zasileń bazy przez nowe dane. Testom zostały także poddane takie elementy rozwiązania jak skrypt w Pythonie i architektura w SSIS, zajmujące się pobieraniem i transformacją danych, ale również ponowne ładowanie danych do narzędzia raportowego Power BI. Na kolejnych stronach przedstawiamy zrzuty ekranu wraz z krótkimi podpisami, której fazy dotyczyły dane testy. Naturalnie na niektórych z nich widać, że zdarzały się sytuacje, w których niektóre wiersze nie dawały załadować się do hurtowni. Błędy te wynikały między innymi z nieokreślonych wartości w danych komórkach lub powtarzaniem się niektórych wierszy, ale konsekwentnie były obsługiwane przez zademonstrowane rozwiązanie.

9.1 Skrypt Pythonowy generujący pliki CSV

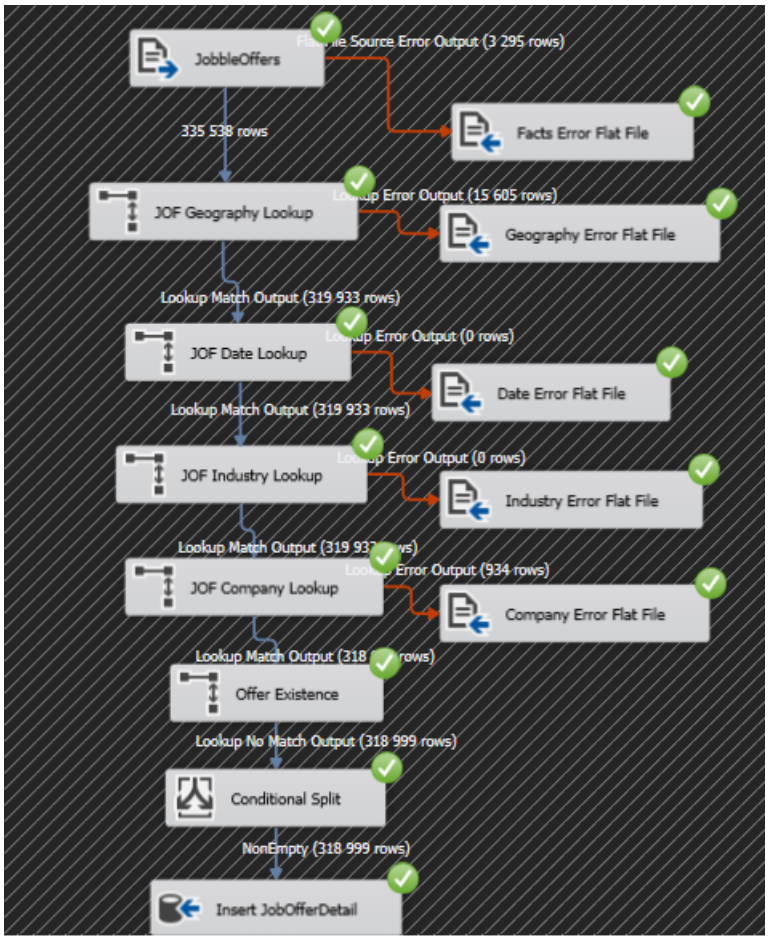


Rysunek 11: Skrypt Python

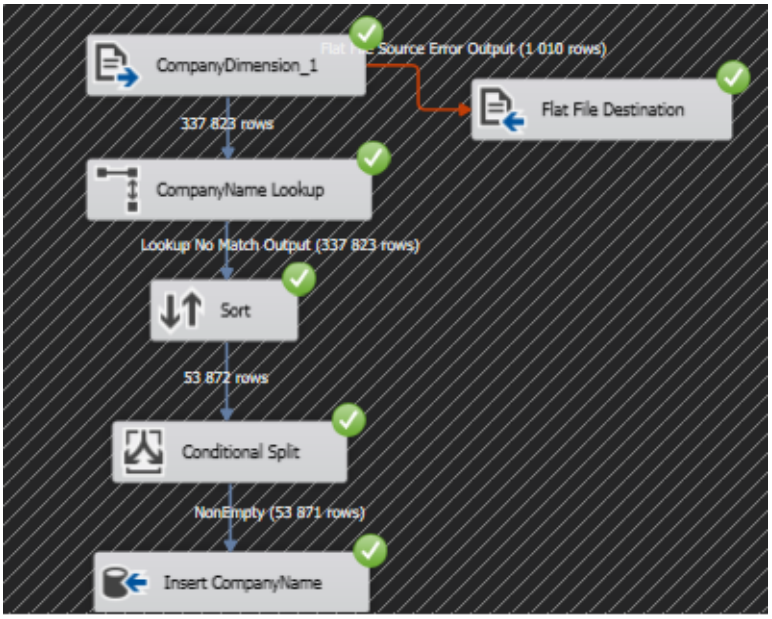
9.2 Pierwsze ładowanie danych do hurtowni



Rysunek 12: Schemat zasilenia hurtowni danymi

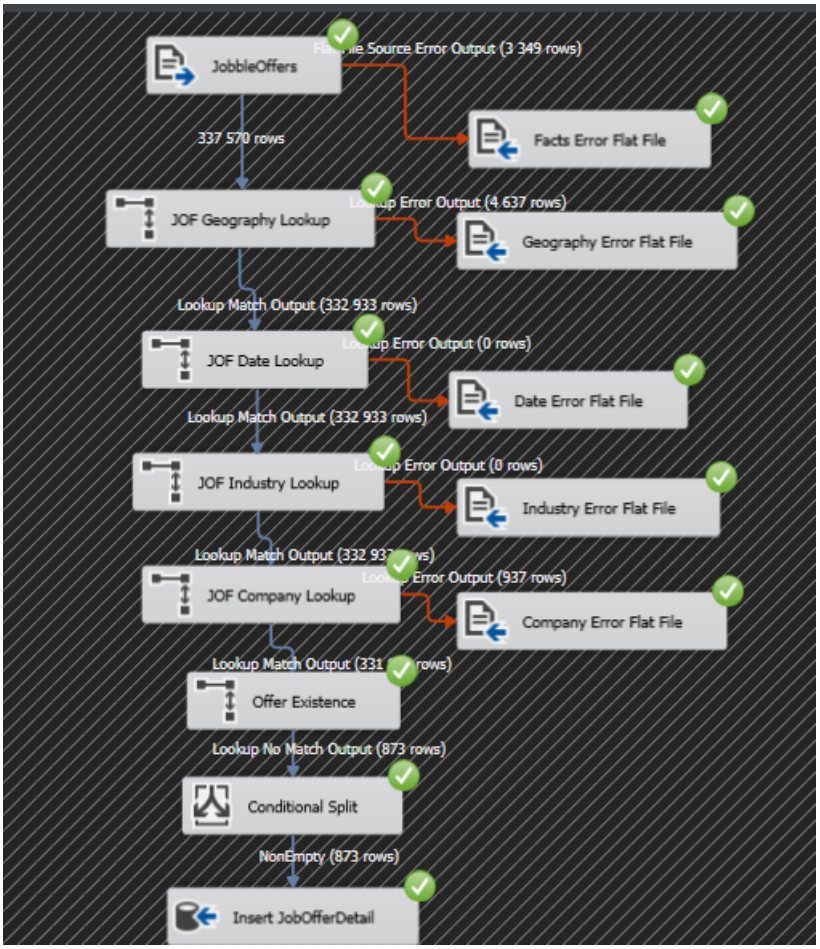


Rysunek 13: Ładowanie tabeli faktów o ofertach

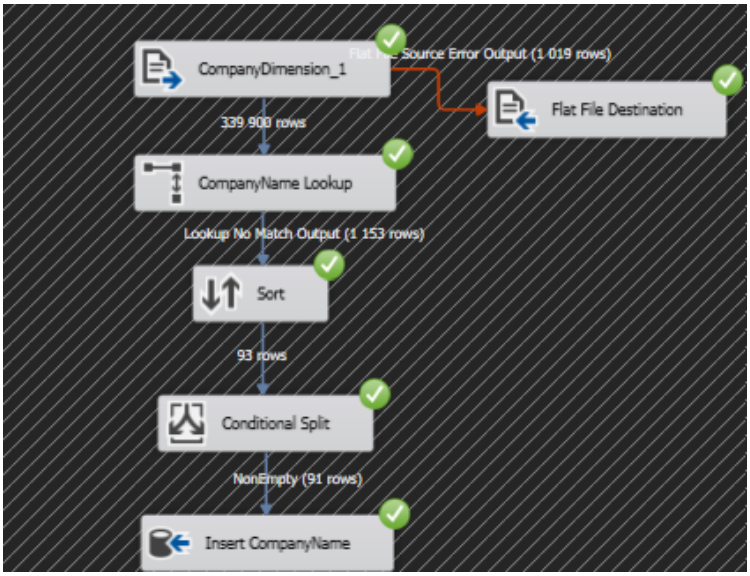


Rysunek 14: Ładowanie tabeli wymiarów przechowującej dane o firmach publikujących ogłoszenia

9.3 Zasilenie hurtowni nowymi danymi

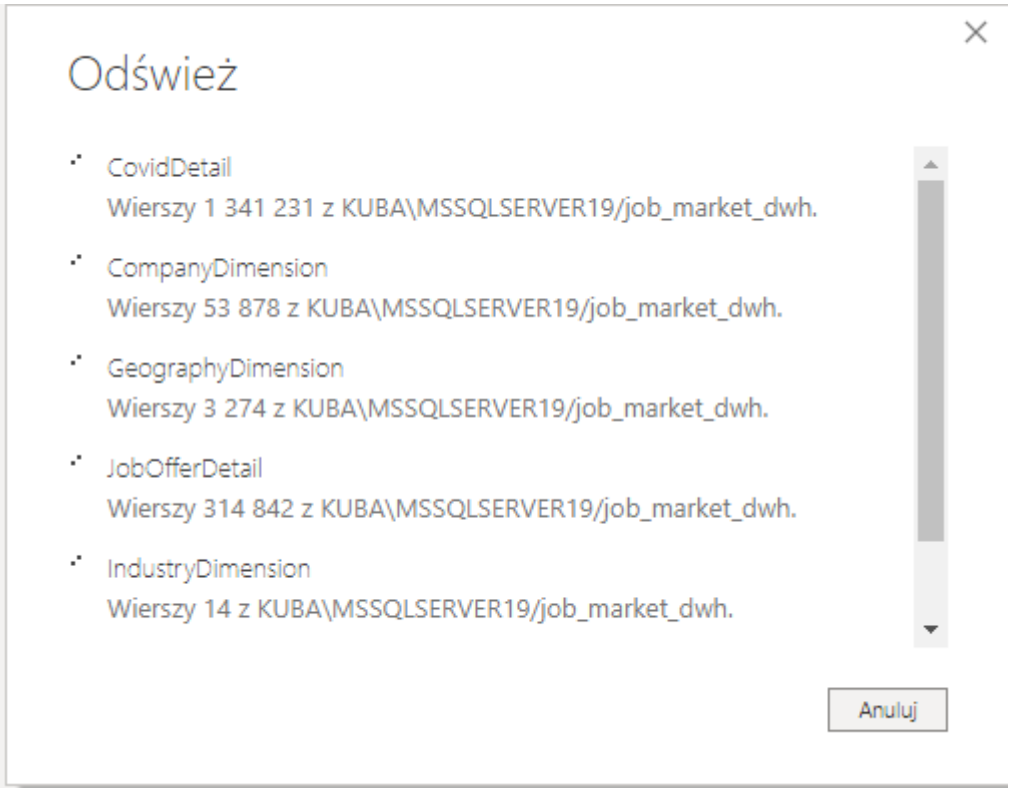


Rysunek 15: Zasilenie tabeli faktów o ofertach nowymi danymi



Rysunek 16: Zasilenie tabeli wymiarów przechowującej dane o firmach publikujących ogłoszenia nowymi danymi

9.4 Ładowanie danych do narzędzia PowerBI



Rysunek 17: Ładowanie danych do narzędzia PowerBI

10 Opis podziału pracy w zespole

Z racji faktu, iż wszystkie etapy projektu realizowane były wspólnie w stałej komunikacji wszystkich członków zespołu, w poniższej rozpisce do każdego etapu przypisana została osoba, która włożyła w dany etap najwięcej pracy, oraz był on realizowany na jej lokalnym sprzęcie.

- Pobieranie danych z API oraz stworzenie skryptu w języku Python - Mikołaj Jakubowski
- Faza ładowania danych do hurtowni z wykorzystaniem SSIS - Elżbieta Jowik
- Tworzenie raportu z wykorzystaniem narzędzia PowerBI - Ada Gąssowska
- Dokumentacja projektu - wspólnie.