Maciej Jacheć (189967)

**Raport z projektu zaliczeniowego na przedmiot Hurtownia Danych**

**Cel projektu**

Celem projektu jest stworzenie Hurtowni Danych w modelu płatka śniegu, implementacja diagramu na serwer SQL Management Studio, pokazanie funkcjonalności oprogramowania VisualStudio poprzez implementację danych z FlatFile, Excel oraz OLE DB. Dodatkowo, pokazana zostanie funkcjonalność narzędzia VisualStudio w postaci kostki OLAP oraz jej przykładowy przegląd. Żeby przegląd się udał zostanie również przeprowadzone zasilenie poglądowymi rekordami tabel zawierających się w diagramie.

**Zaplanowanie diagramu**

Diagram w postaci płatka śniegu ma prezentować dane dotyczące dokumentów księgowych. Tabela faktów poza kluczami głównymi i obcymi będzie zawierać miarę w postaci wysokości kwoty na dokumencie.

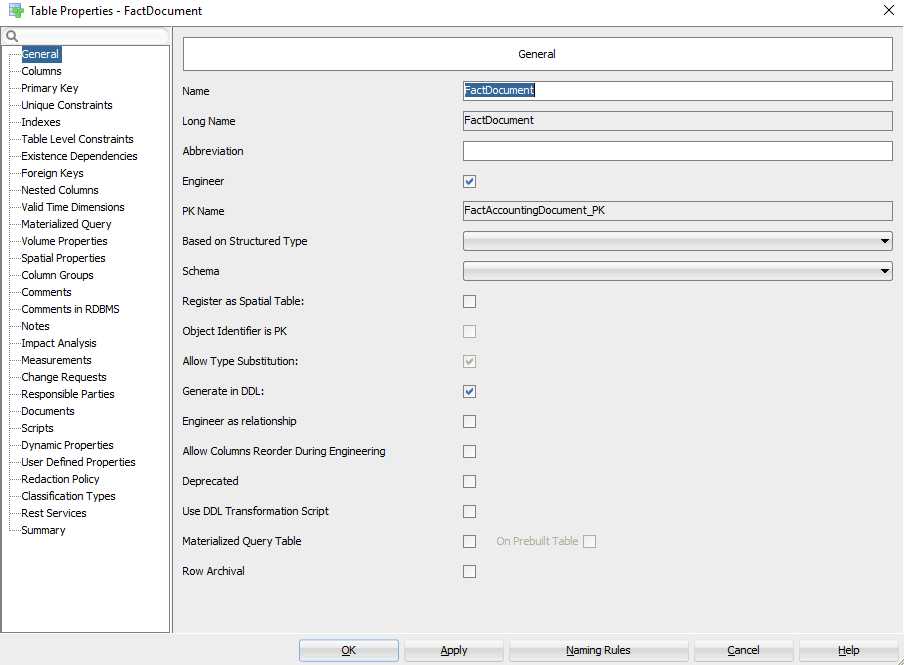
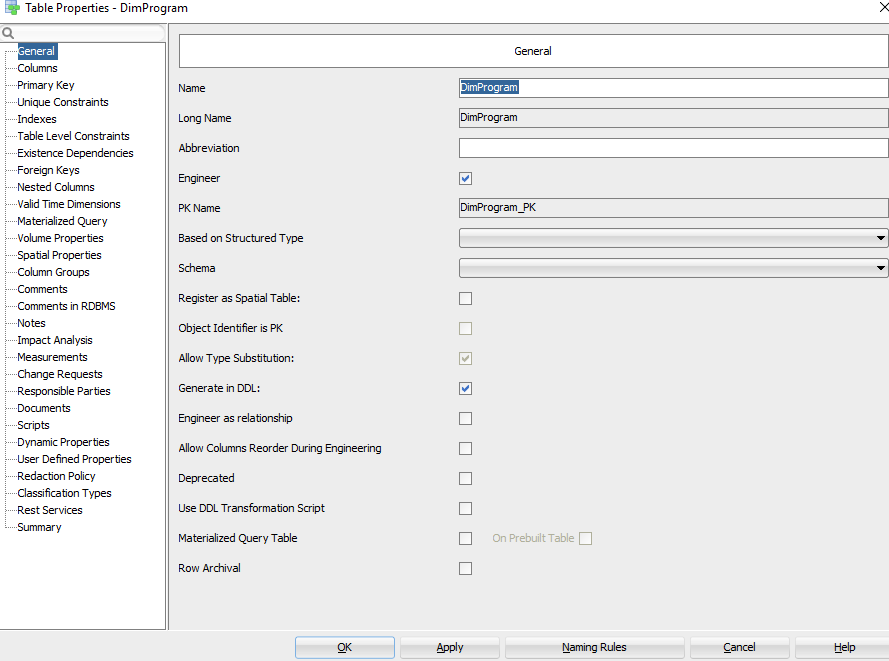
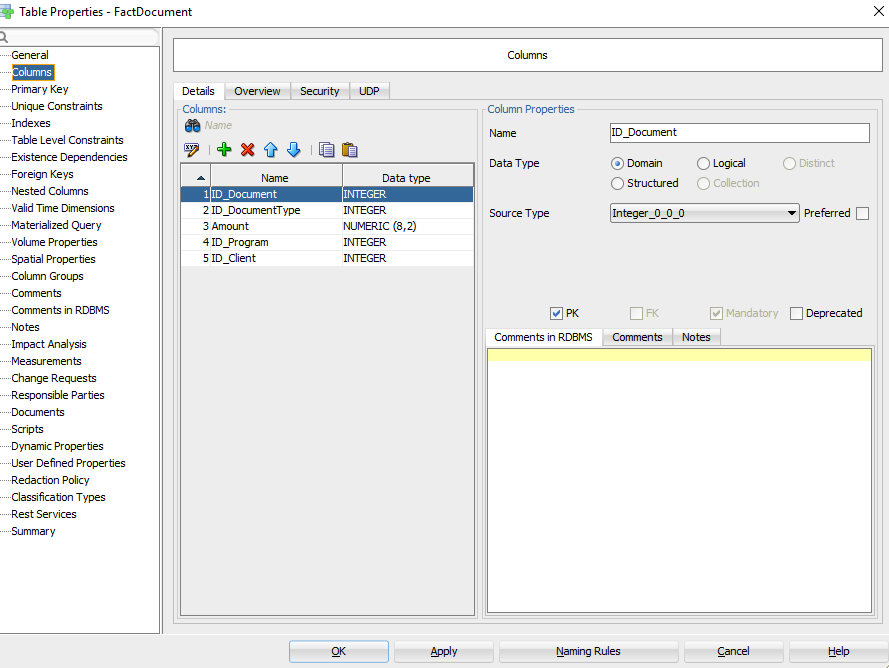
Tabele wymiarów będą przechowywać informację o:

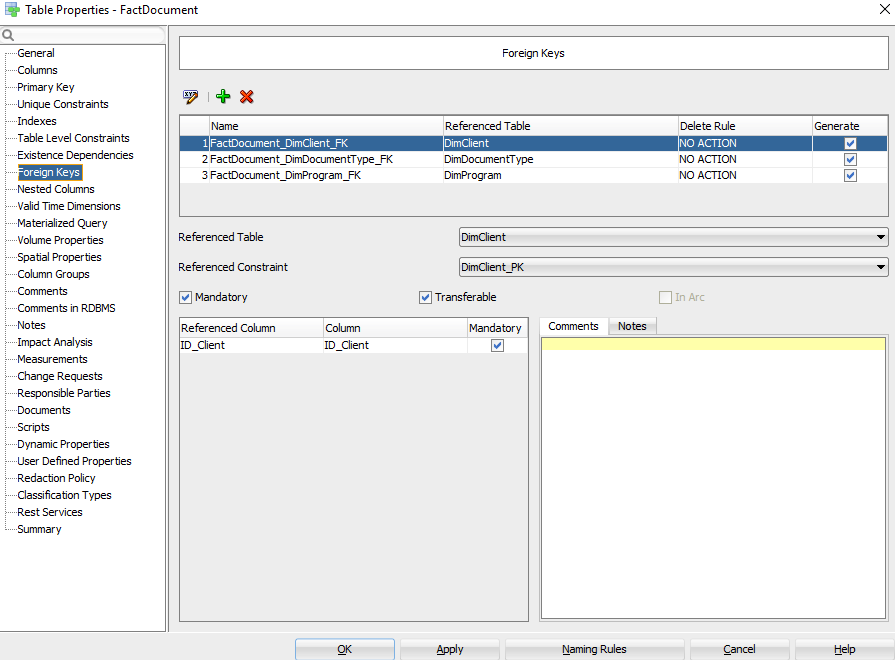
* Programie, w którym przechowywany jest dokument (atrybuty to Nazwa programu, Data zakończenia dostępu do programu)
* Typie dokumentu (atrybuty to rodzaj dokumentu, rejestr księgowy, oraz numer)
* Kontrahencie, który wystawił albo dla którego została wystawiona faktura (atrybuty to nazwa klienta, forma prawna działalności prowadzonej przez klienta oraz rok założenia)

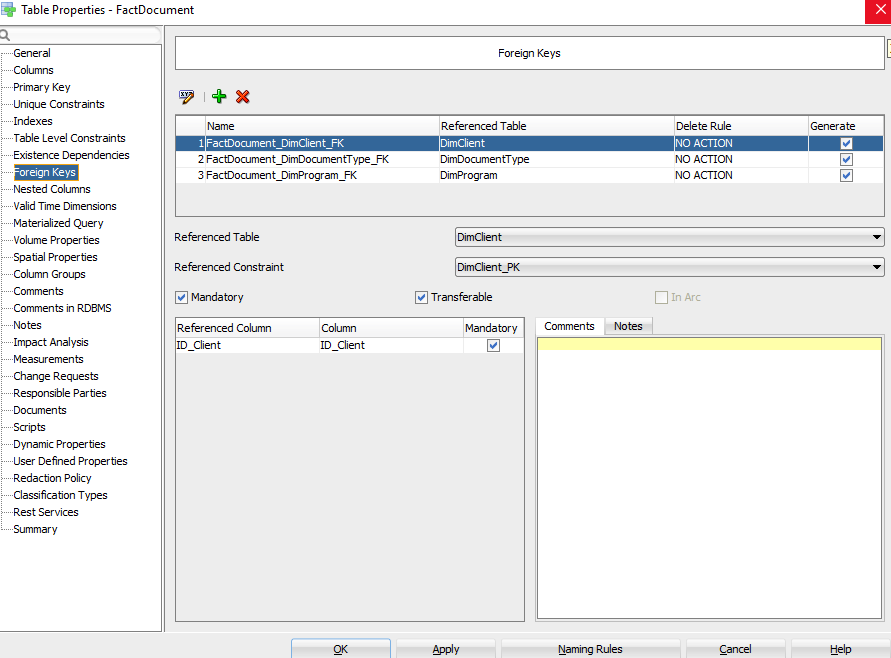
Jako że diagram ma postać płatka śniegu to tabela wymiaru dotycząca programu będzie zawierać klucz obcy dla tabeli wymiarów przechowującej dane o dziale biura rachunkowego który w dane oprogramowanie jest zaopatrzony. Jej atrybutami będą: szef działu, pokój działu, numer wewnętrzny oraz nazwa pracownika, który z oprogramowania korzysta.

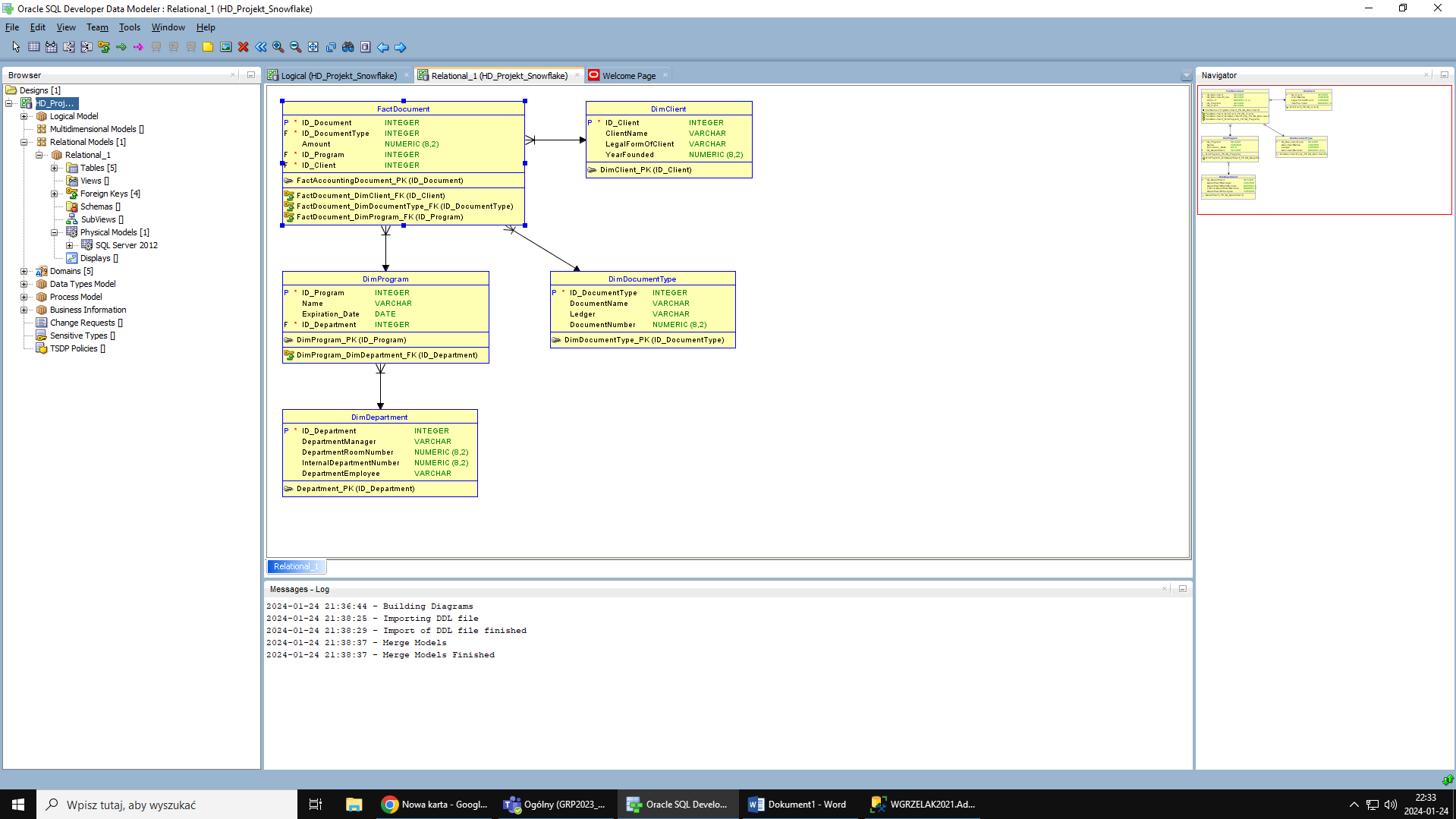
W ten sposób utworzona zostaje niewielkich rozmiarów Hurtownia Danych, która pozwala uzyskać wszystkie najważniejsze informację o dokumencie księgowym z którego może korzystać zarówno przedsiębiorca jak i biuro rachunkowe które prowadzi księgowość ów przedsiębiorcy.

1. **Stworzenie modelu płatka śniegu w oprogramowaniu Oracle SQL Developer Data Modeler:**

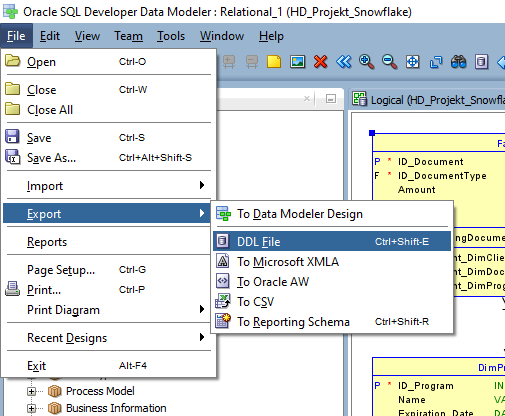
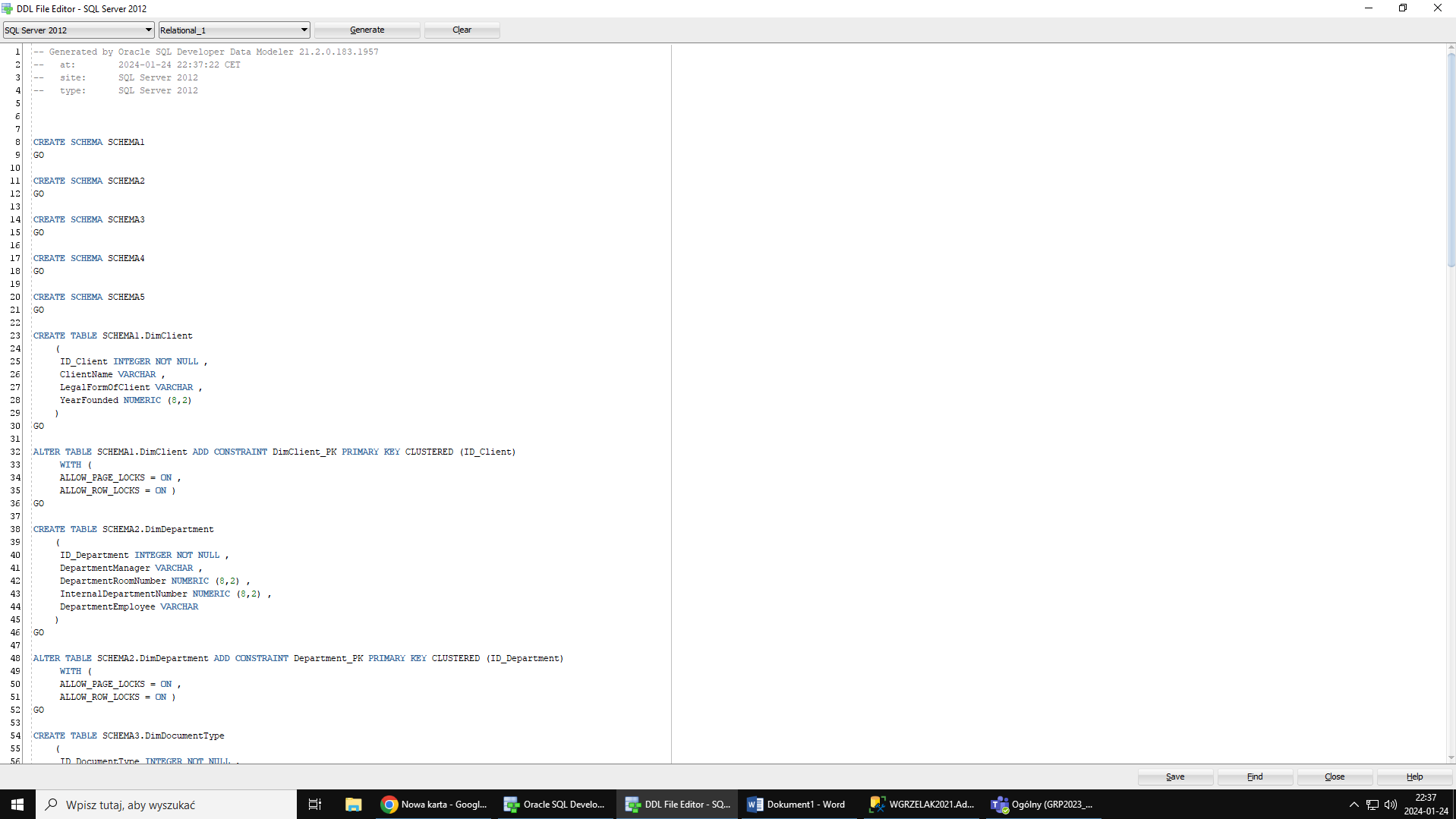
* ****Dodanie tabel faktów i wymiarów i odpowiednie nazewnictwo:
* ****Dodanie kolumn dla atrybutów, miar, kluczy głównych i obcych, określenie typów danych przechowywanych w kolumnach, oznaczenia kluczy głównych:

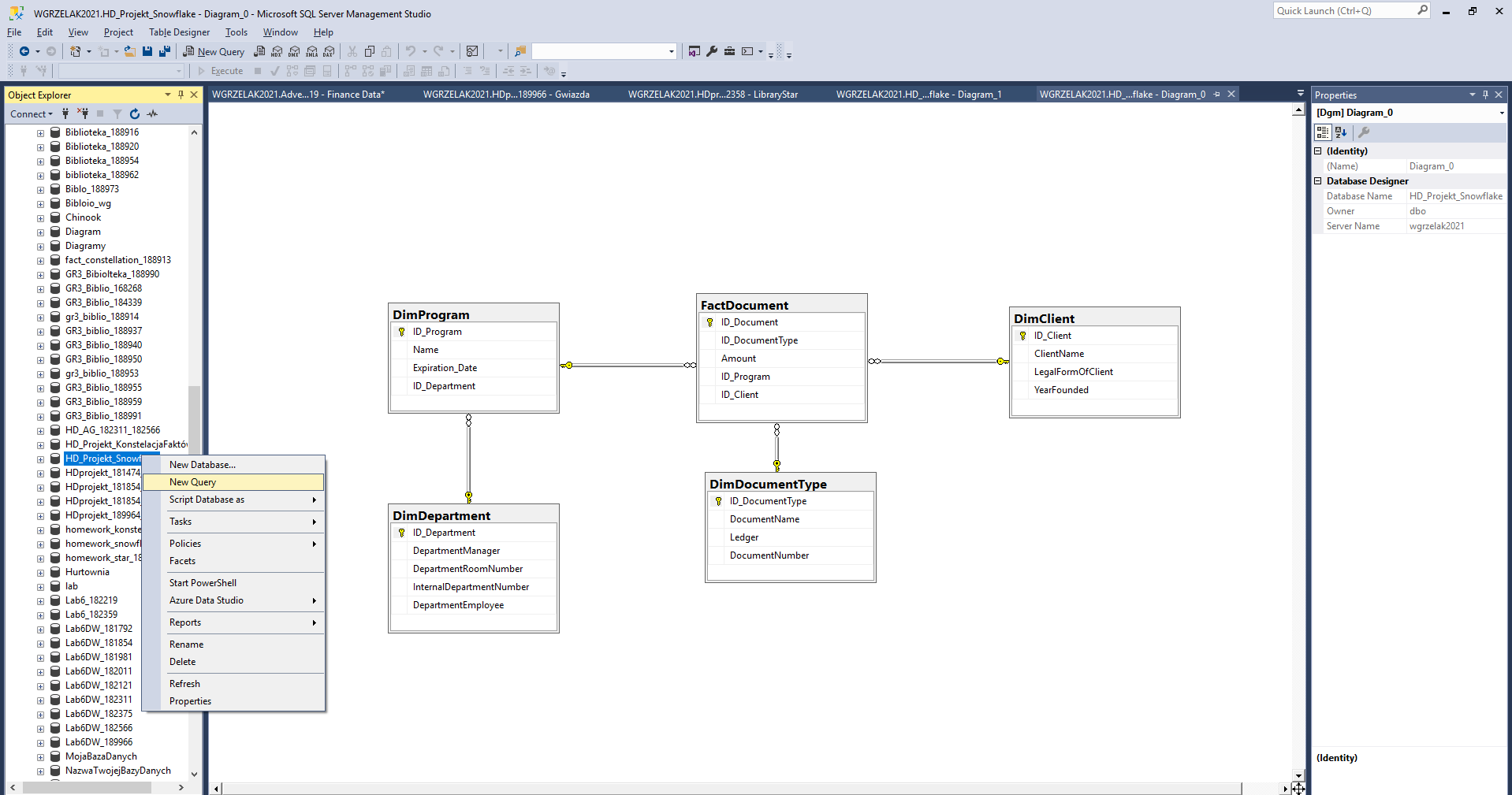


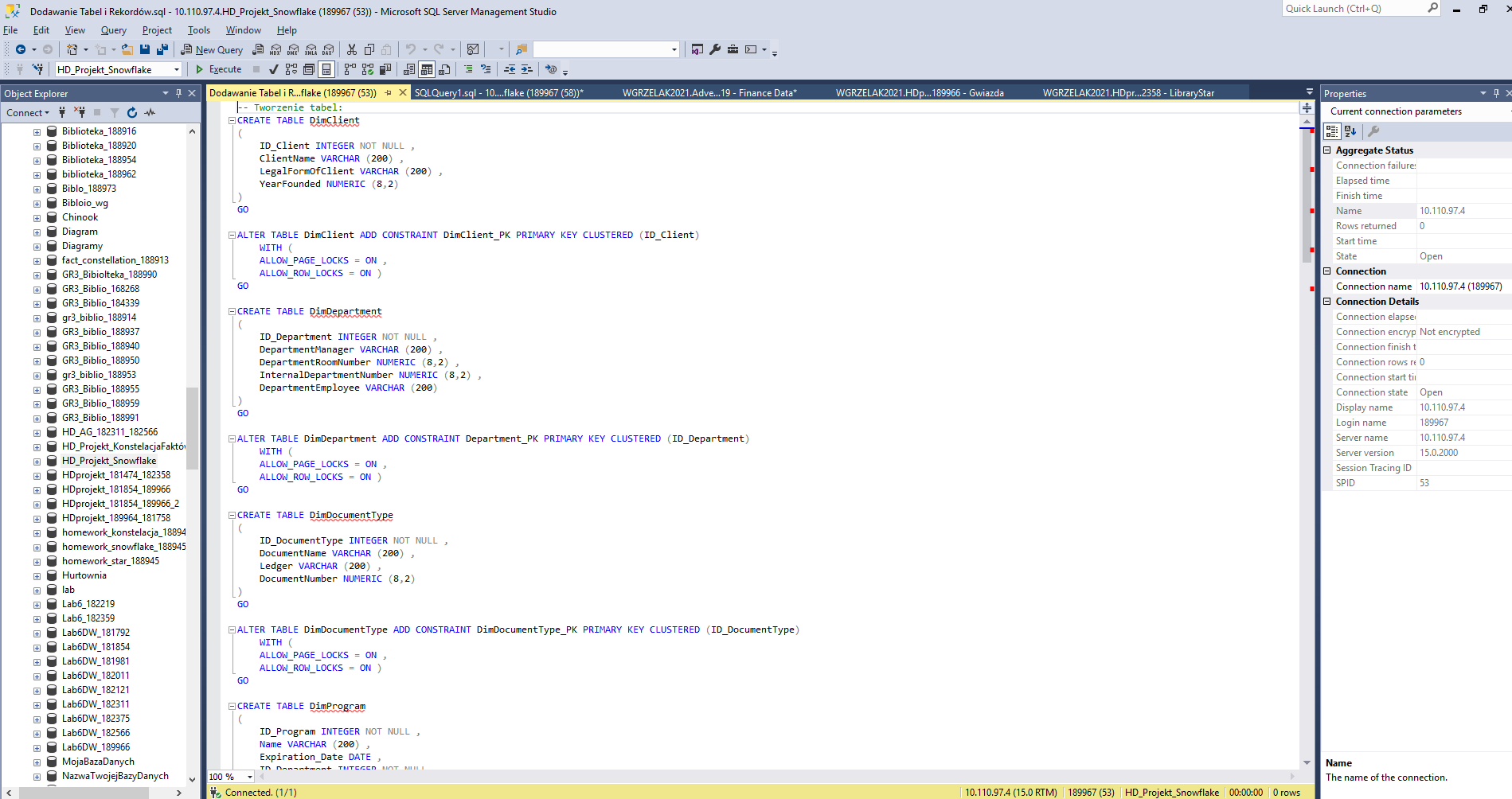
* Dodanie kluczy obcych, odpowiednie ułożenie tabel w celu uzyskania modelu płatka śniegu:



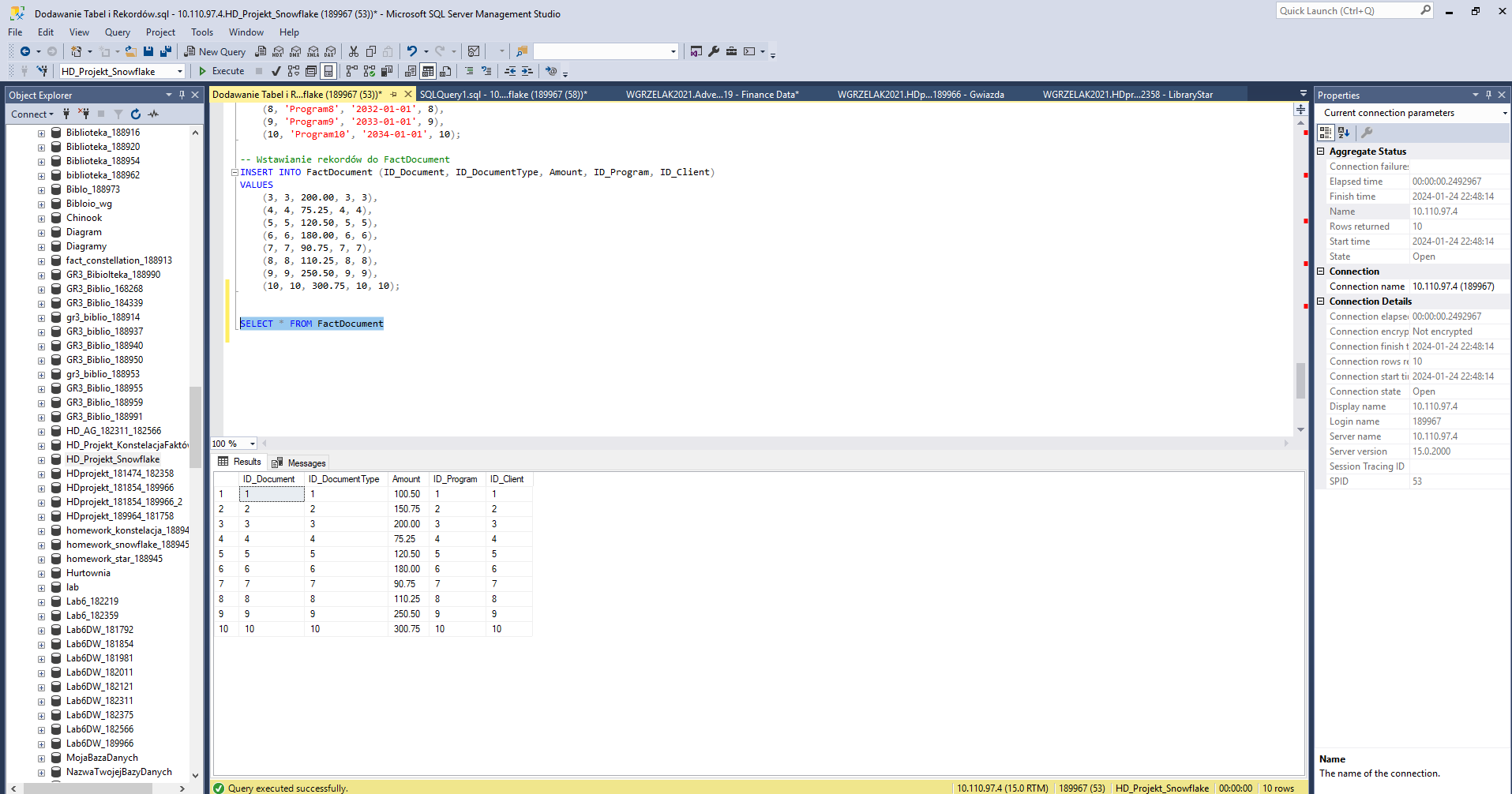
* Wygenerowanie pliku DDL w celu skopiowania skryptu i zaimplementowanie diagramu do BazyDanych SQL Server Management Studio:



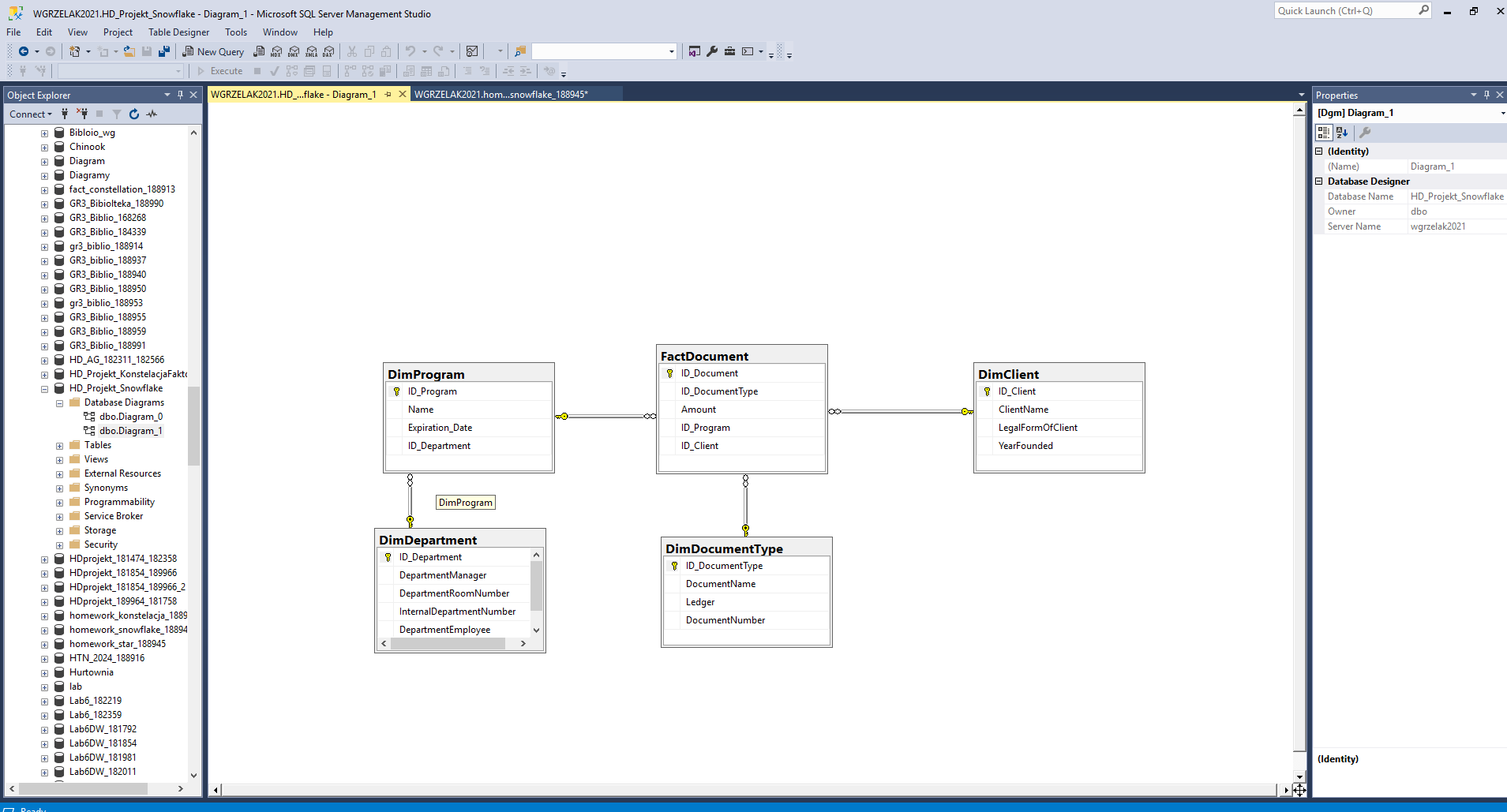
* Dodanie okna nowego zapytania do już wcześniej utworzonej bazy danych na potrzeby projektu HD\_Projekt\_Snowflake:
* Dodawanie odpowiednio zmodyfikowanego skryptu wygenerowanego z SQL Oracle Developer DataModeler (usunięcie zapisów „SCHEMA”) oraz napisanie kodu, który ma na celu dodanie 10 przykładowych rekordów do kolumn w tabelach:



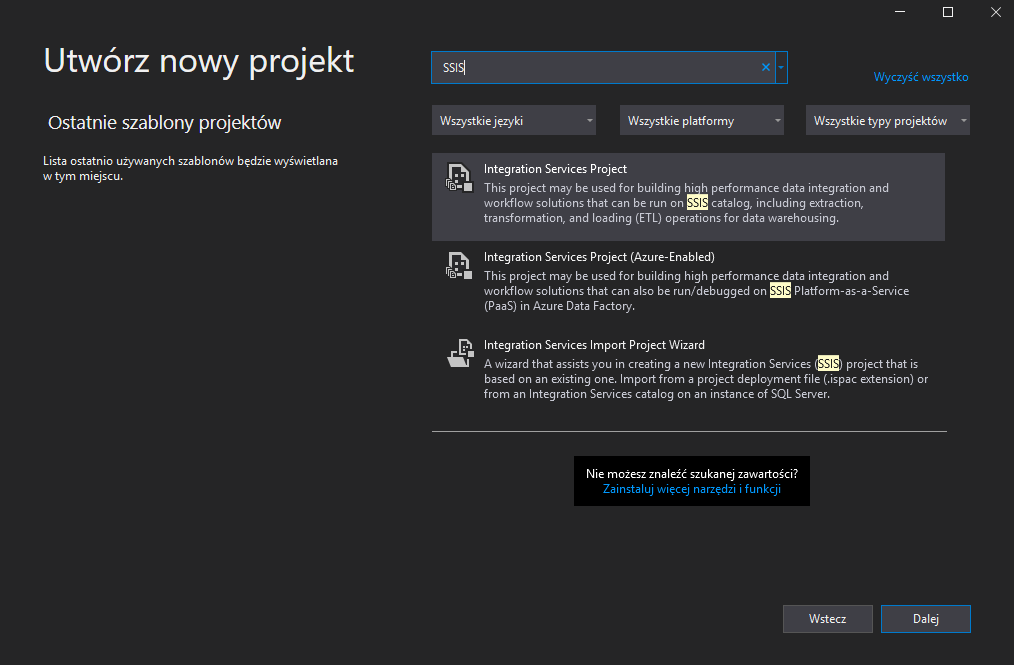
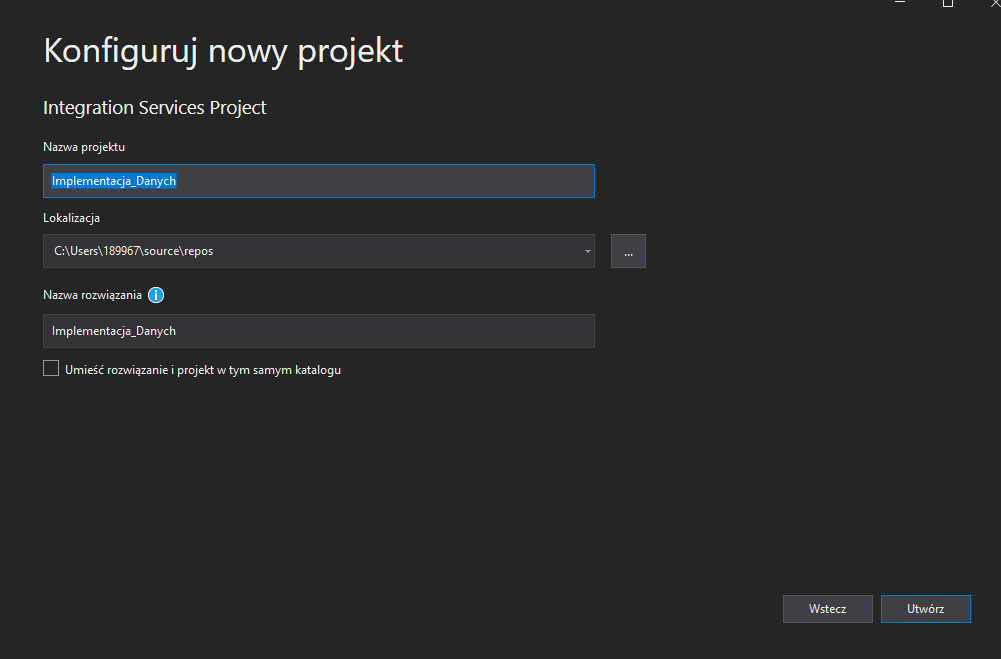
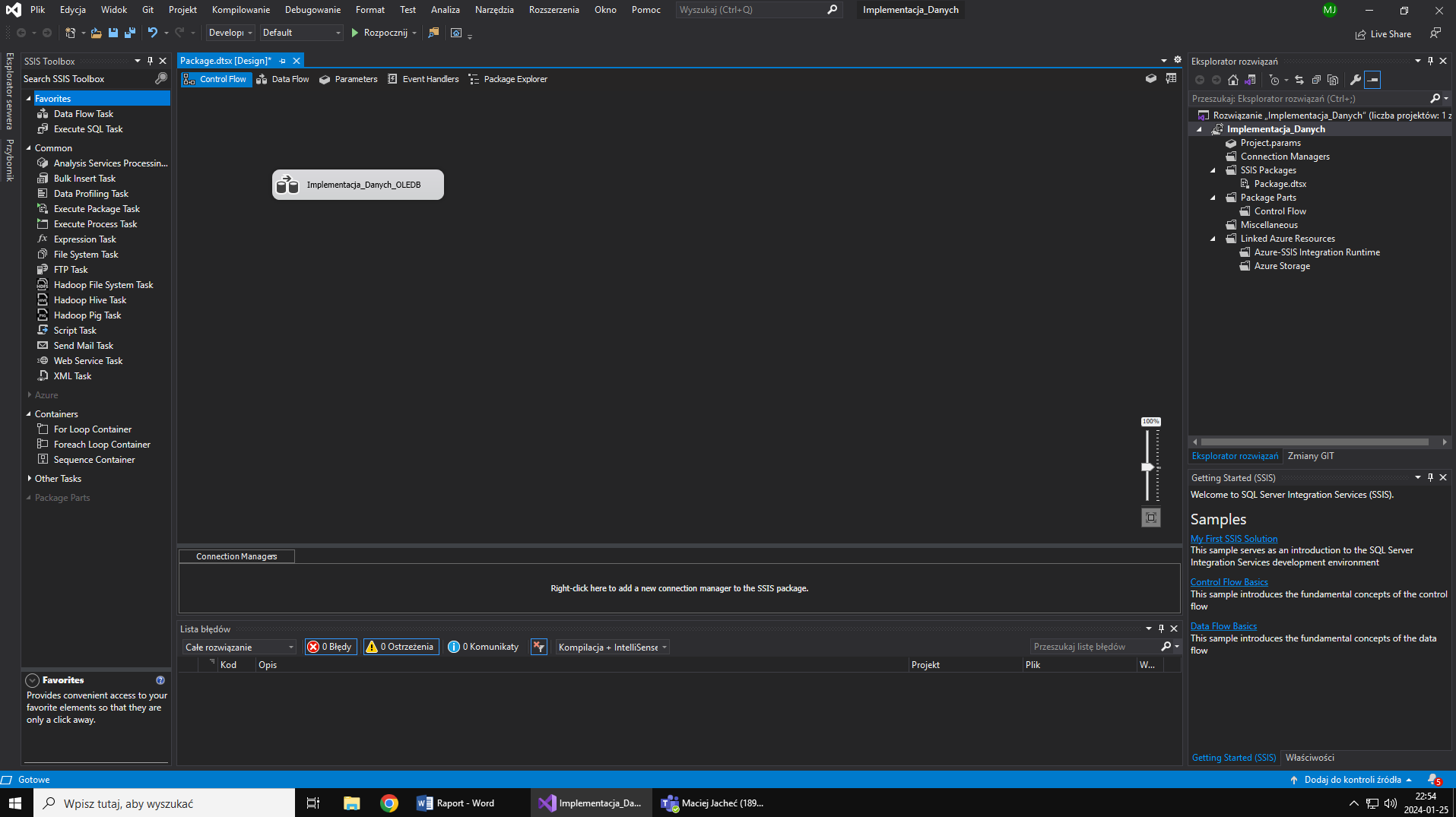
* Sprawdzenie:

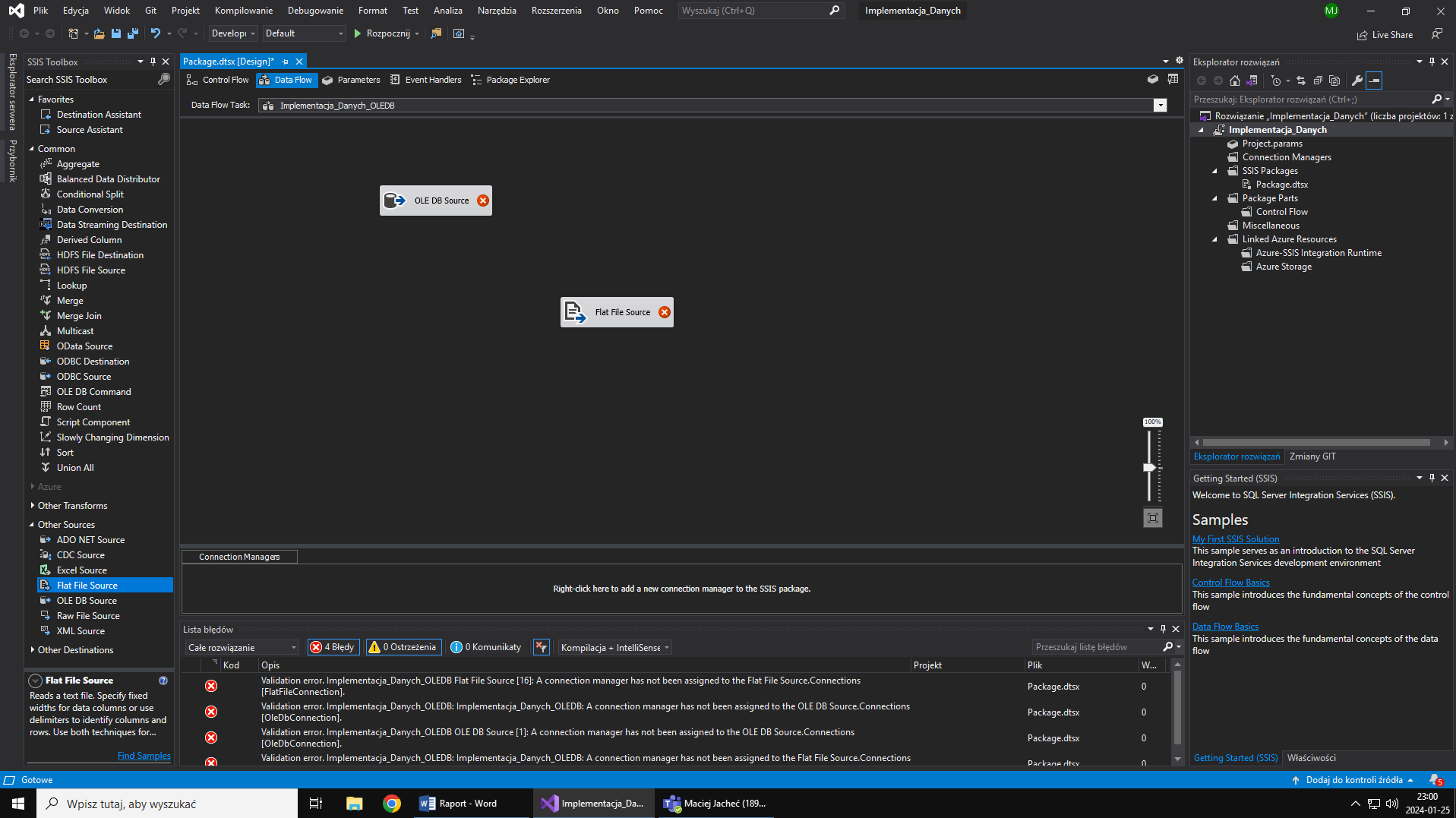


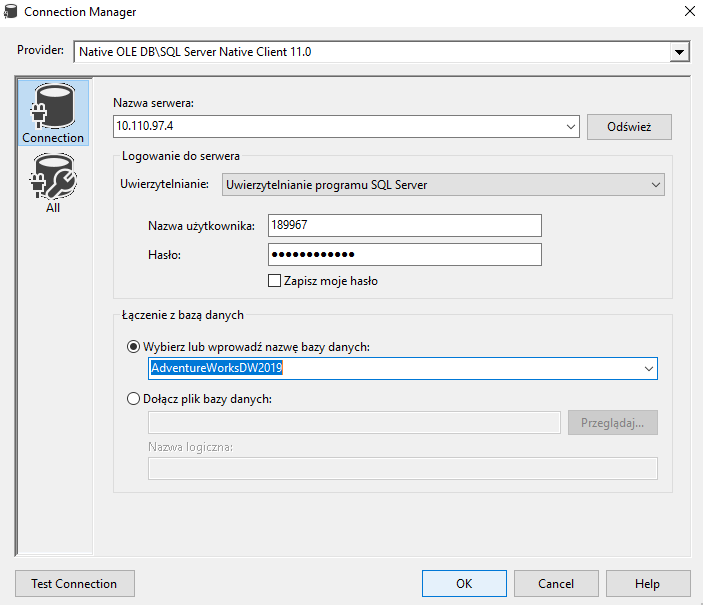
Skoro bez błędów wyświetlają się rekordy tabeli faktów to oznacza to, że dane w pozostałych tabelach również zostały dodane prawidłowo. W innym wypadku, jeżeli kolumny kluczy obcych w tabelach wymiarów byłyby błędne, kod SELECT \* FROM FactDocument zaowocowałby błędem co jest bezpośrednio wynikiem charakteru modelu relacyjnego (wszystkie tabele są ze sobą powiązane).

* Utworzenie diagramu w bazie danych:

1. Implementacja danych z FlatFile, Excel oraz OLE DB w VisualStudio

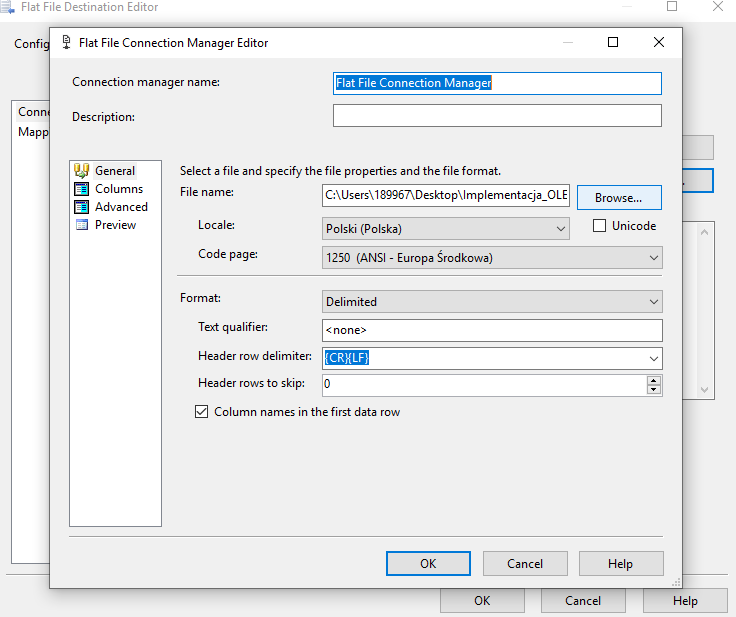
* Uruchomienie programu VisualStudio 2019 i tworzenie nowego projektu. Do implementacji danych można wykorzystać szablon Integration Service Project. Idąc dalej projekt projekt zostaje nazwany „Implementacja\_Danych”.
* W następnym kroku utworzony zostanie nowy Data Flow Task i którego nazwą będzie Implementacja\_Danych\_OLEDB. Dodane zostaje źródło OLE DB Source i miejsce, do którego docelowo zostanie zaimplementowana tabela. W tym wypadku będzie to plik w formacie FlatFile.

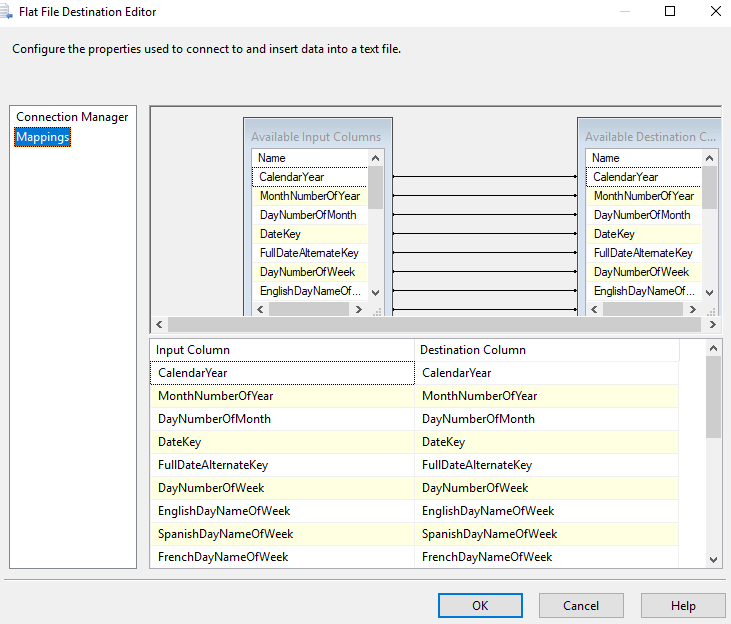


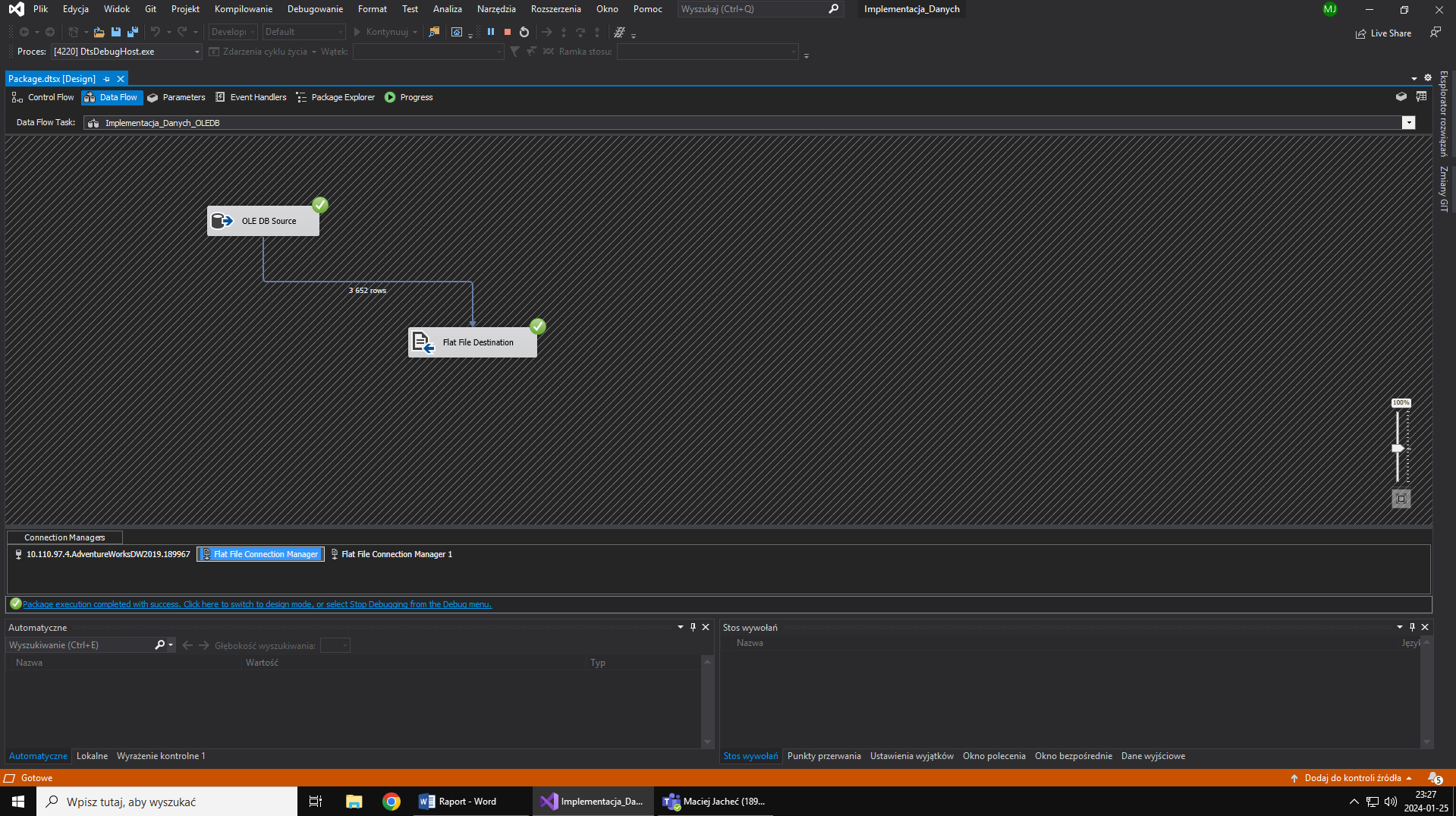
* W OLE DB Source zostaje dodane połączenie z serwerem i wybraną bazą danych z której zostaną „ściągnięte” interesującą nas tabele. Do projektu zostanie wykorzystana BD AdventureWorks2019 z której zostanie „wyciągnięta” tabela DimDate.



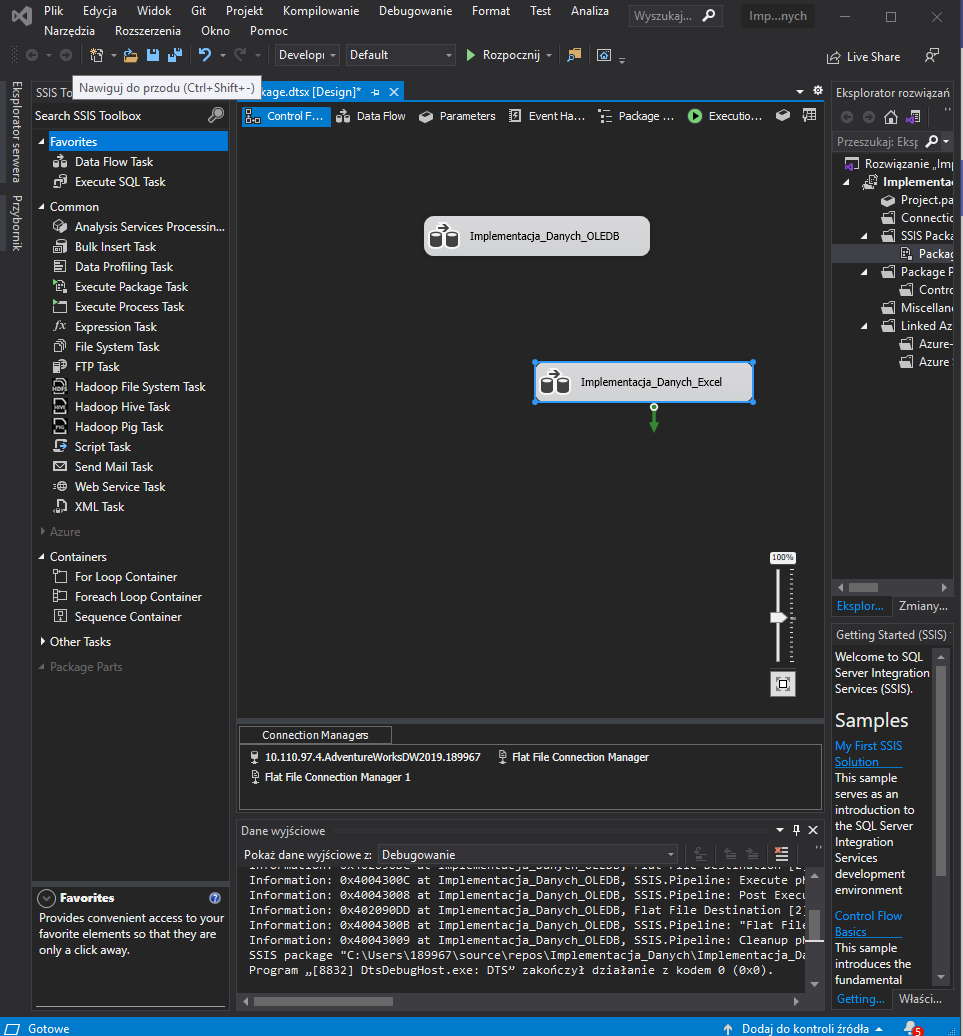
* W kolejnym kroku wykonano łączenie FlatFile Destination z wybranym plikiem tekstowym na komputerze, w którym zostaną zaimplementowane dane. Istnieje możliwość stworzenia nowego dokumentu tekstowego i tak zostało to wykonane w prezentowanym przykładzie. Sprawdzona zostaje poprawność mapowania

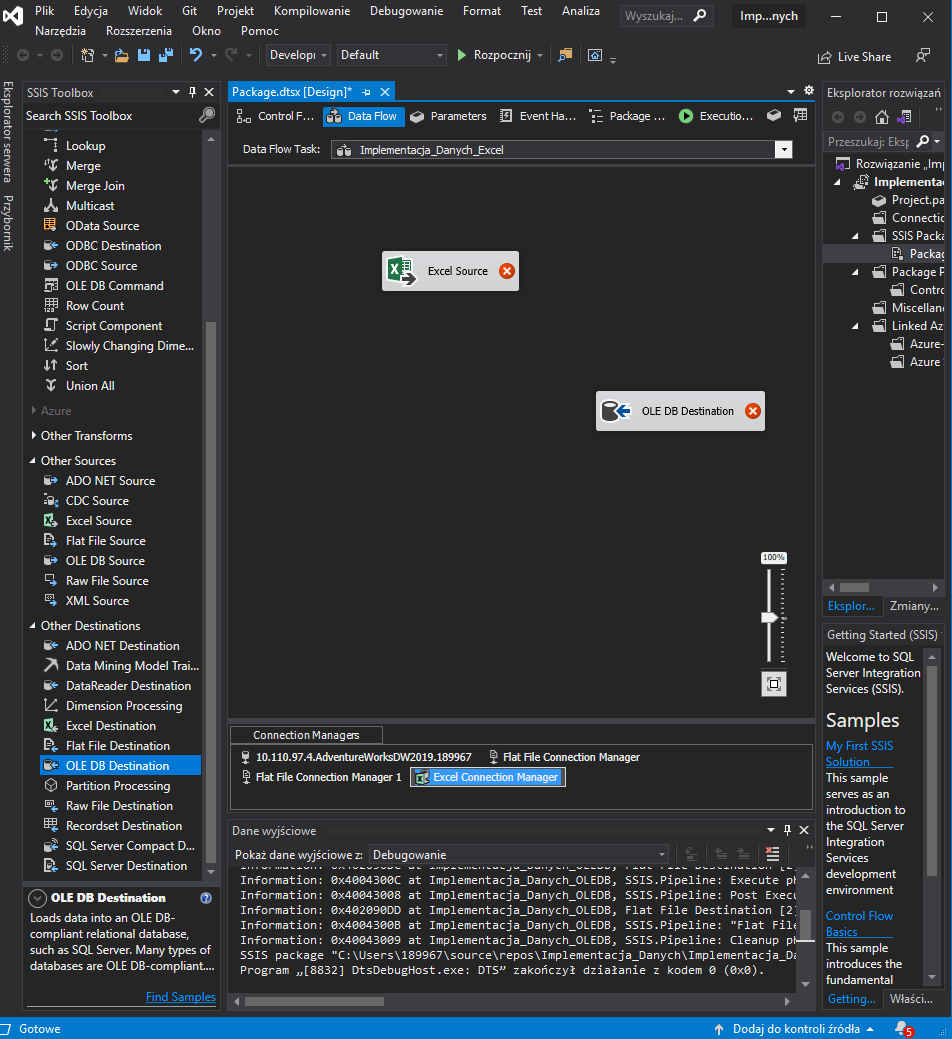




* Na końcu przeprowadzono implementacje. Program nie sygnalizuje żadnych błędów, więc implementacja przebiegła pomyślnie. Sprawdzony zostaje dokument tekstowy. Implementacja przebiegła pomyślnie. Kolejnym krokiem będzie implementacja danych z pliku Excel do OLE DB



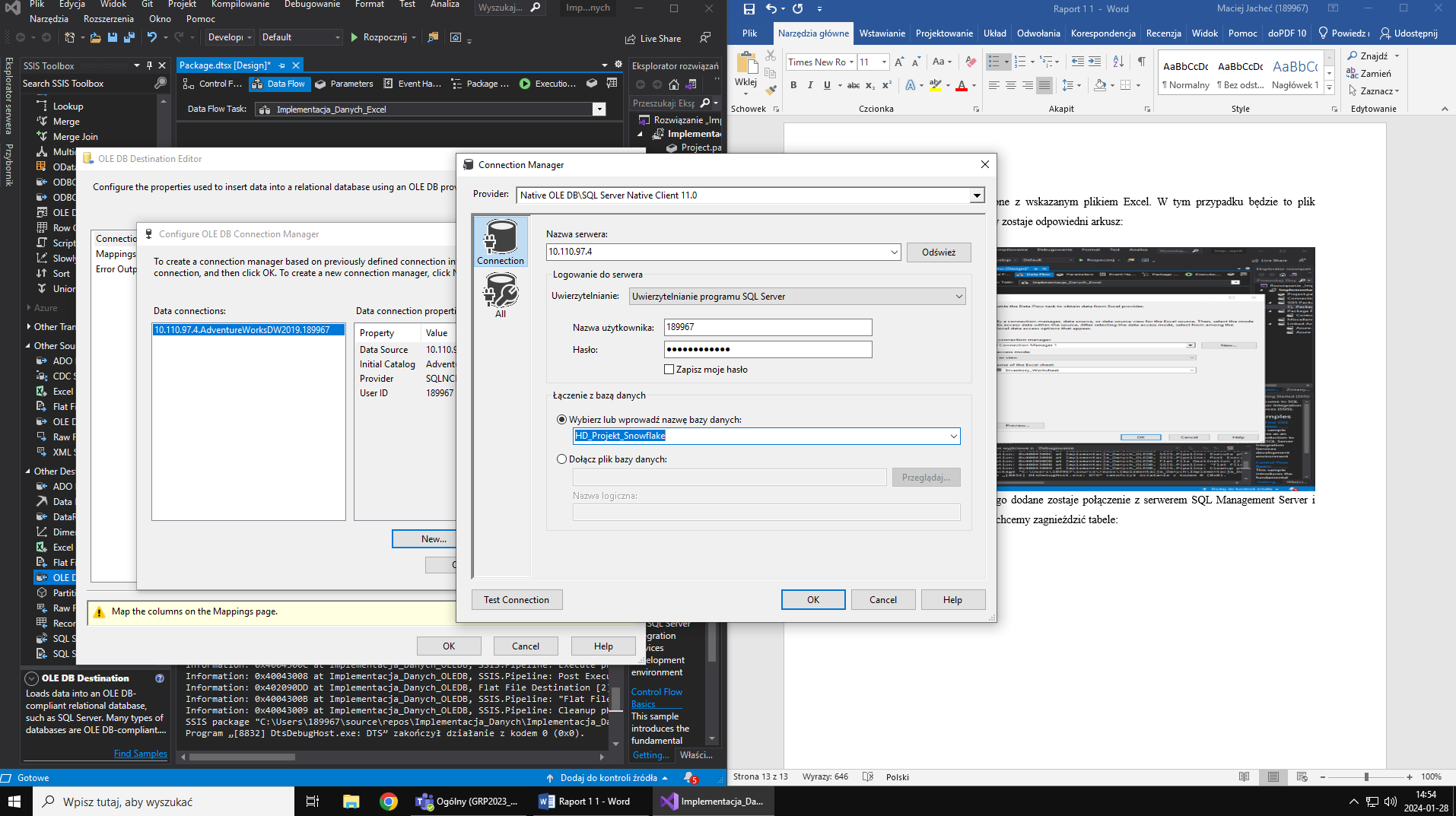
* Utworzony zostaje nowy Data Flow Task który zostanie nazwany Implementacja\_Excel:
* Kolejnym krokiem, podobnie jak w poprzednim przykładzie będzie dodanie źródła i miejsca docelowego z tą różnicą, że w tym przypadku źródłem będzie plik Excel a miejscem docelowym OLEDB:

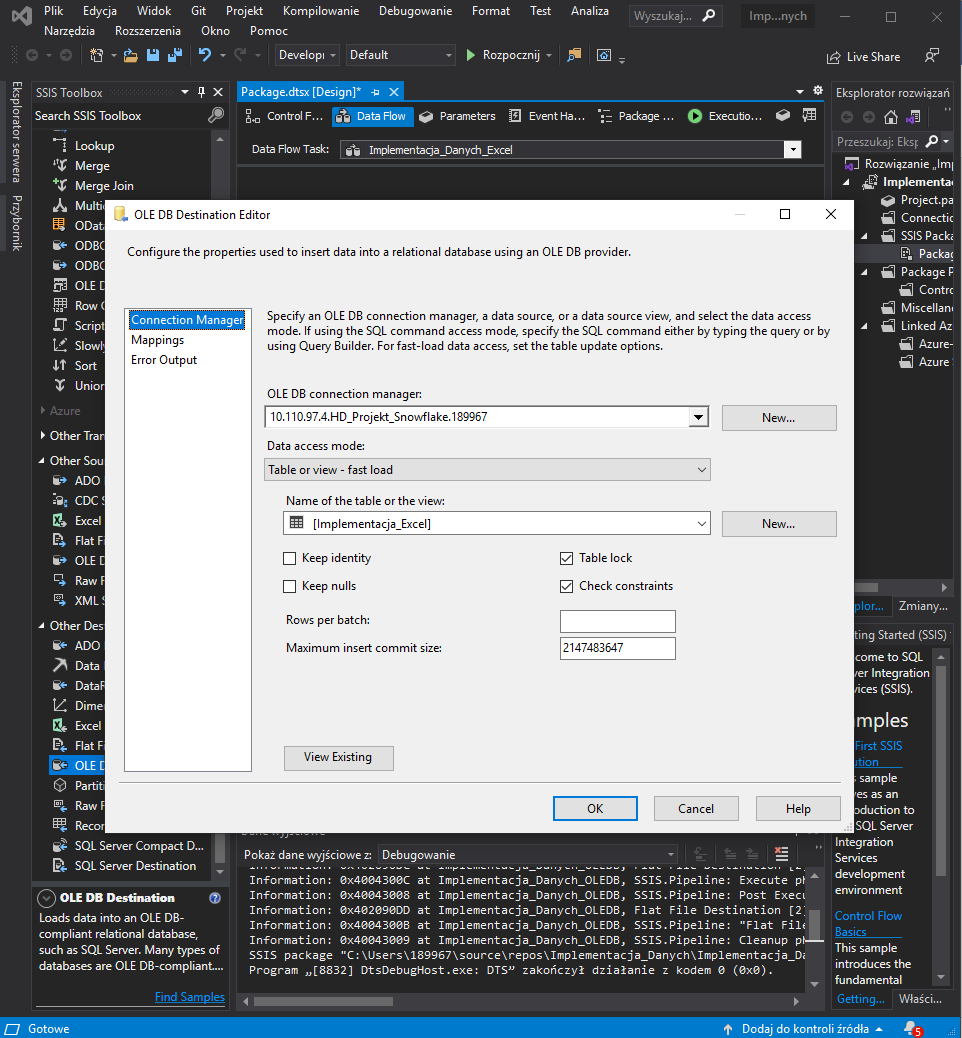
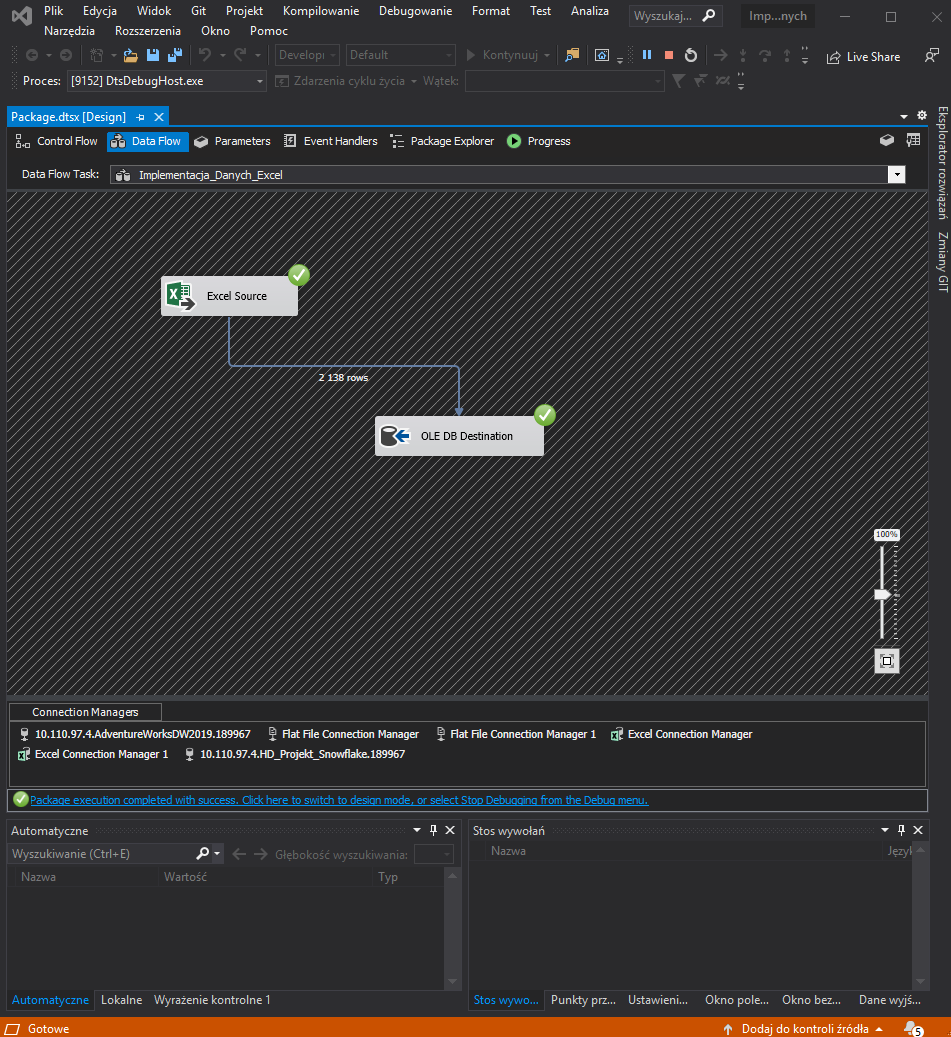


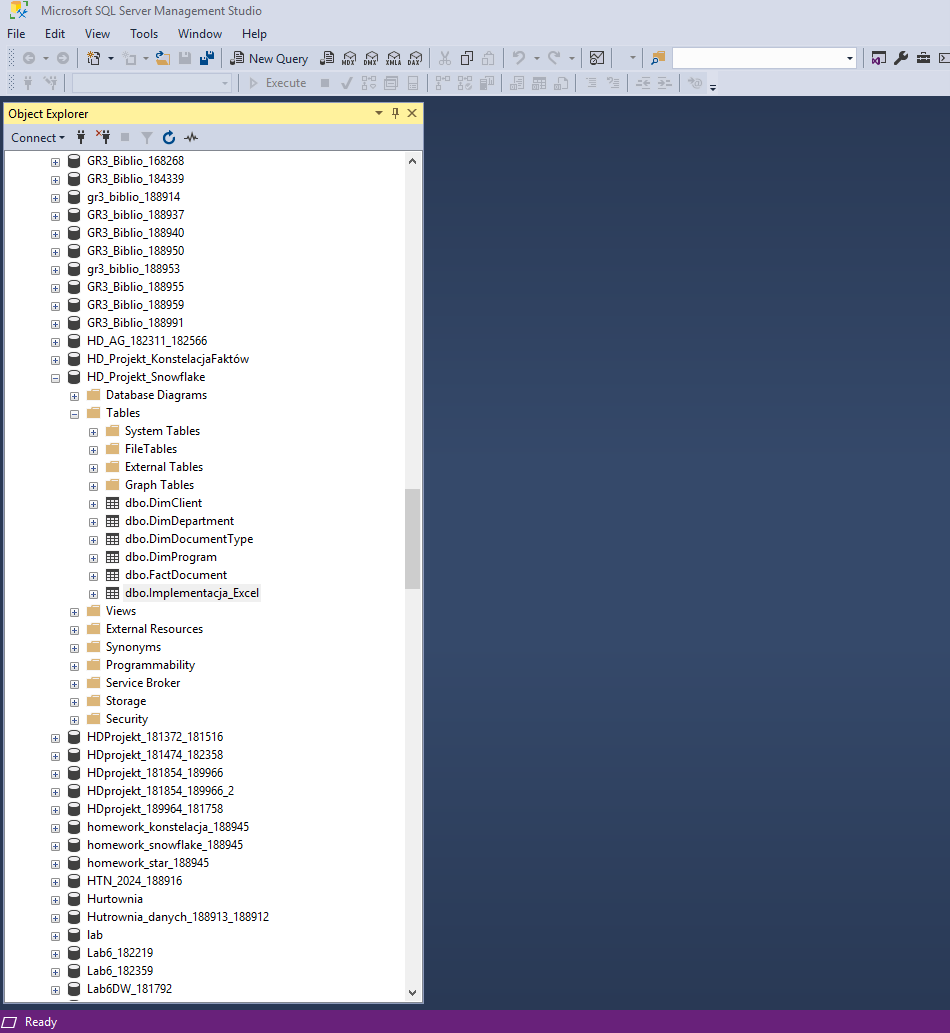
* Źródło zostaje połączone z wskazanym plikiem Excel. W tym przypadku będzie to plik Inventory. Wytypowany zostaje odpowiedni arkusz:

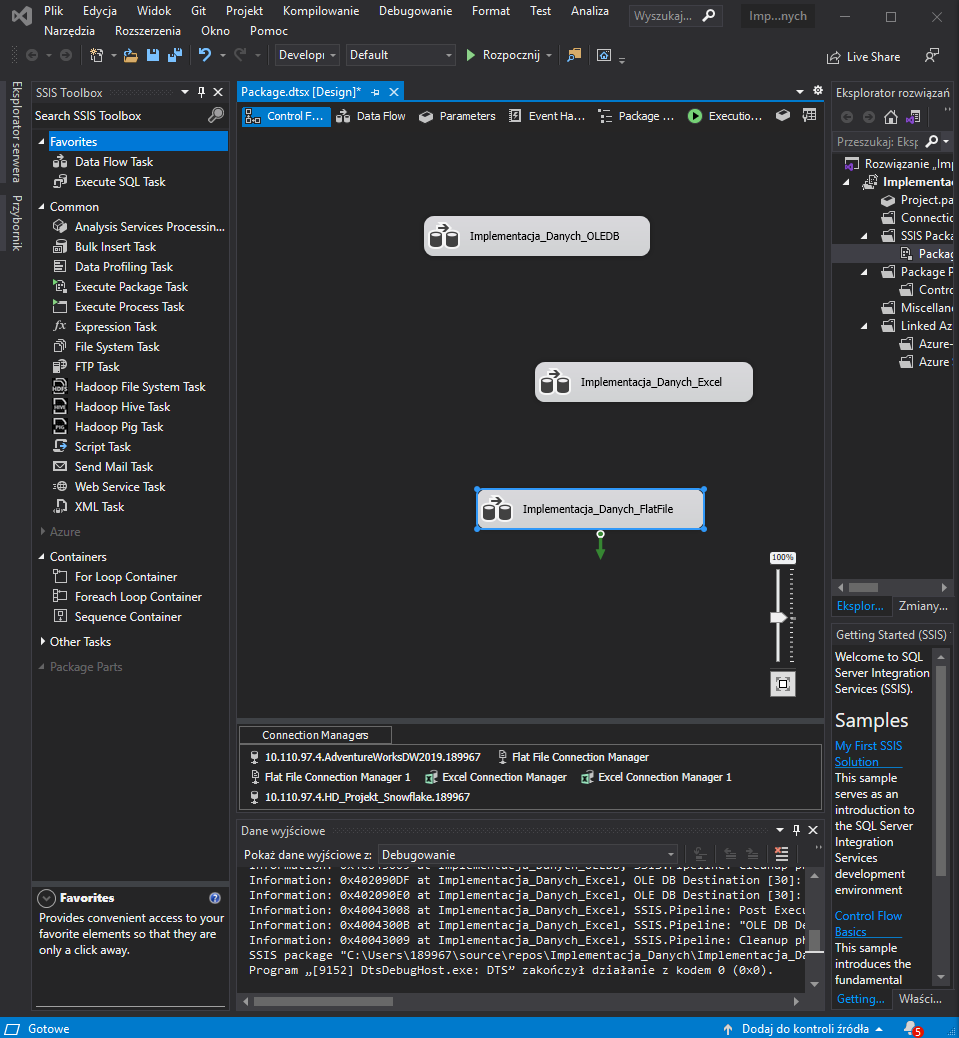


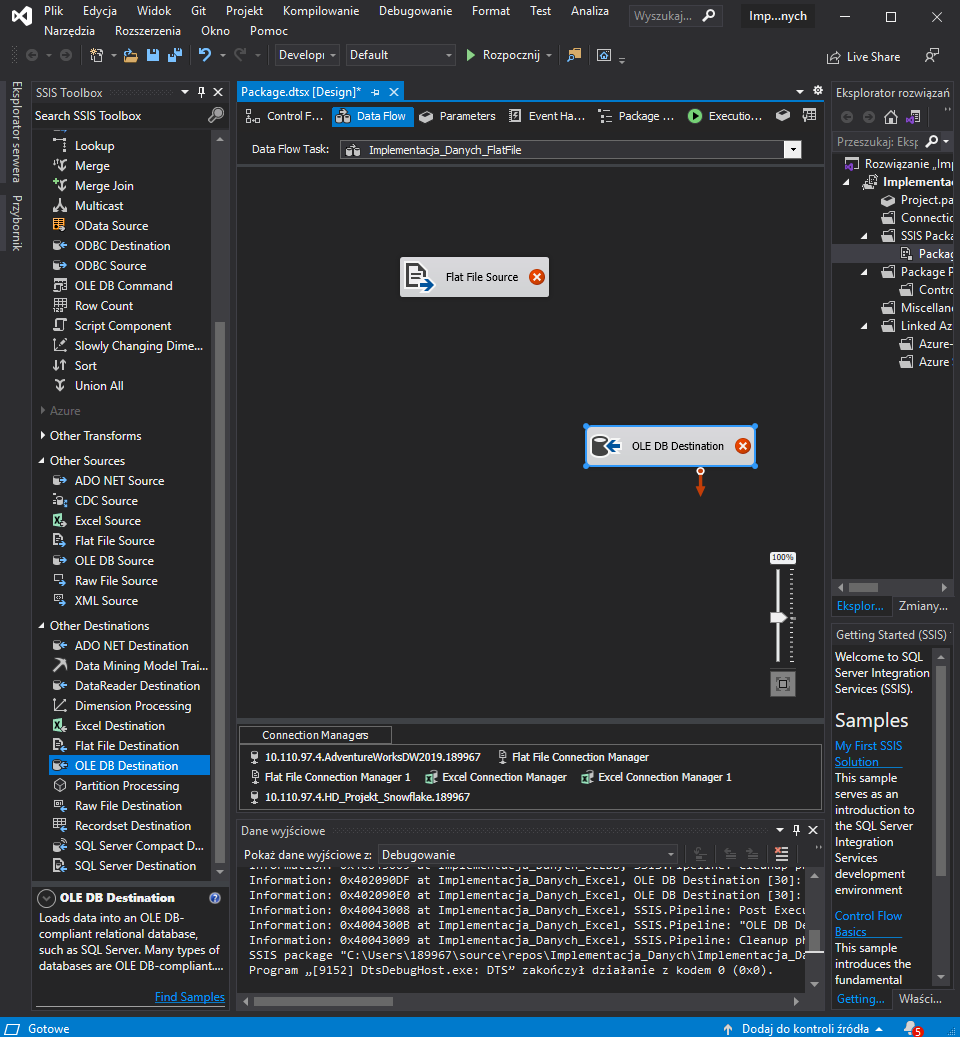
* Dla miejsca docelowego dodane zostaje połączenie z serwerem SQL Management Server i wskazana BD w której chcemy zagnieździć tabele. Na potrzeby projektu została utworzona nowa tabela Implementacja\_Excel:

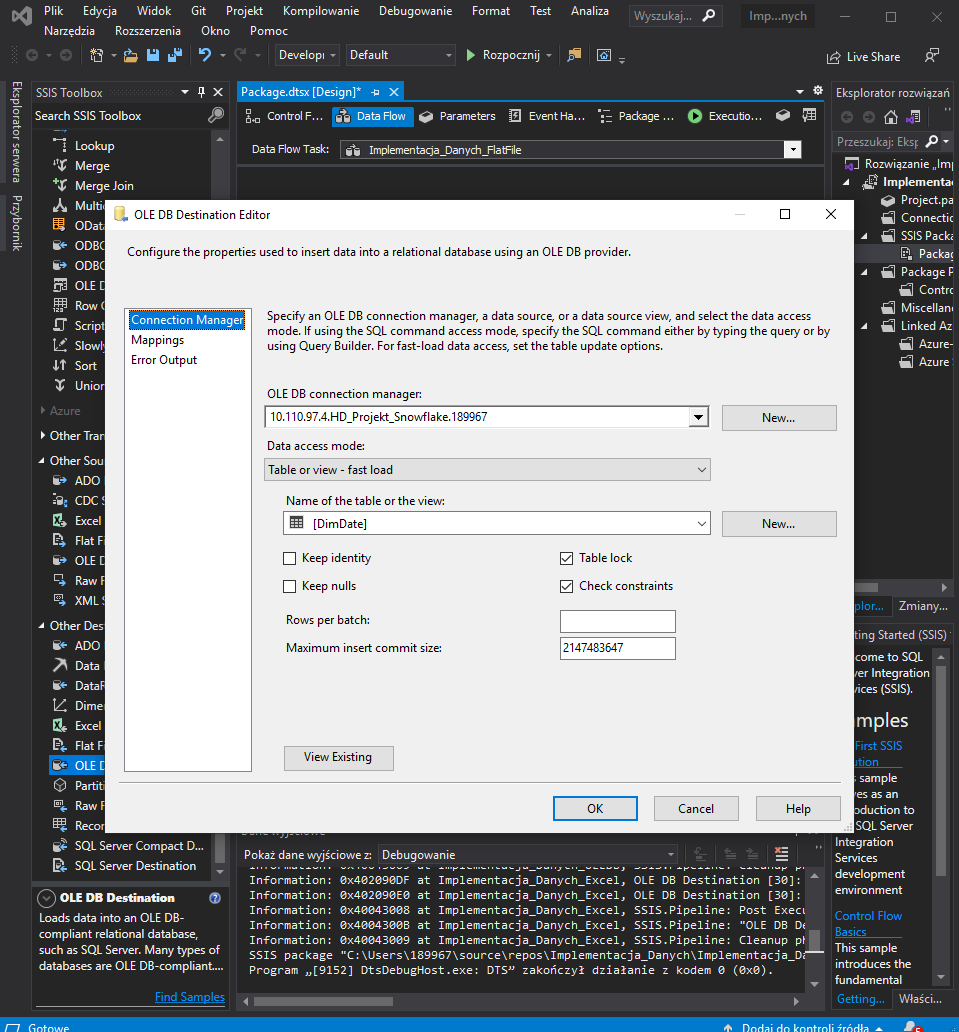
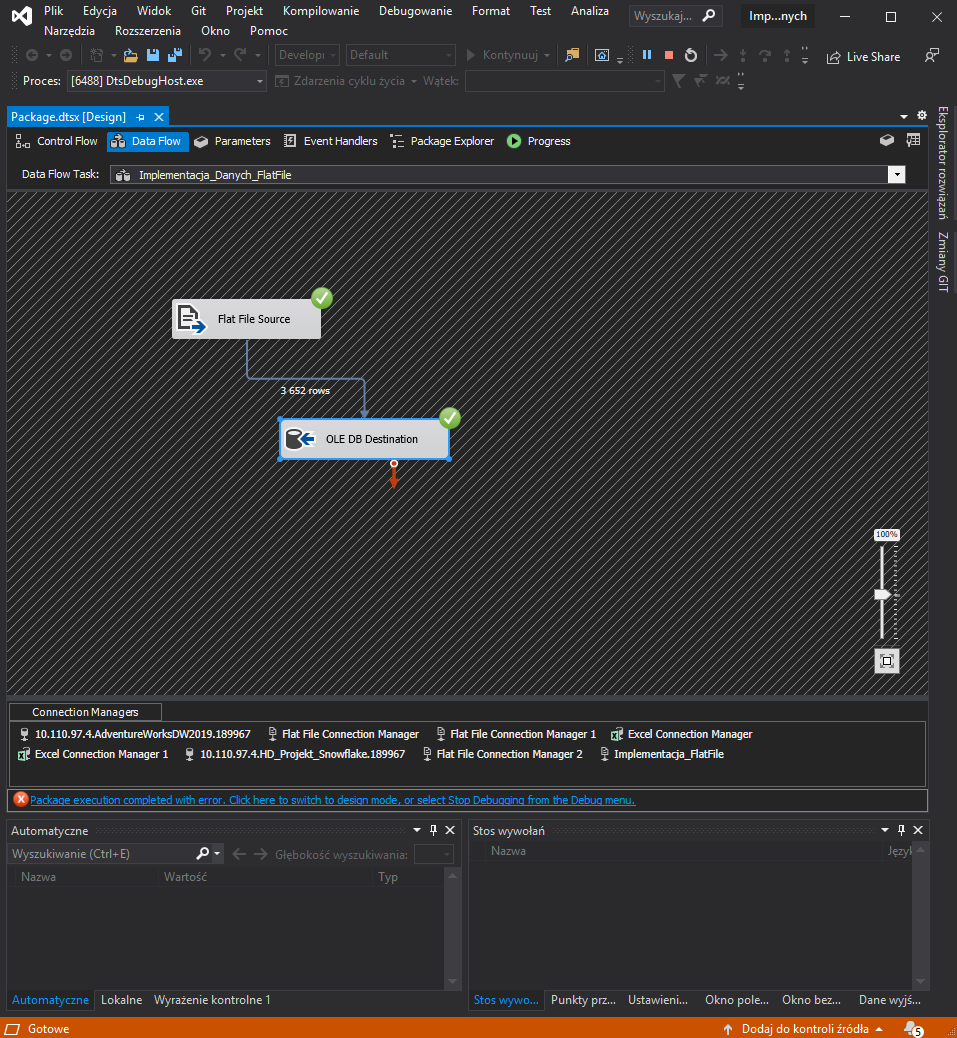
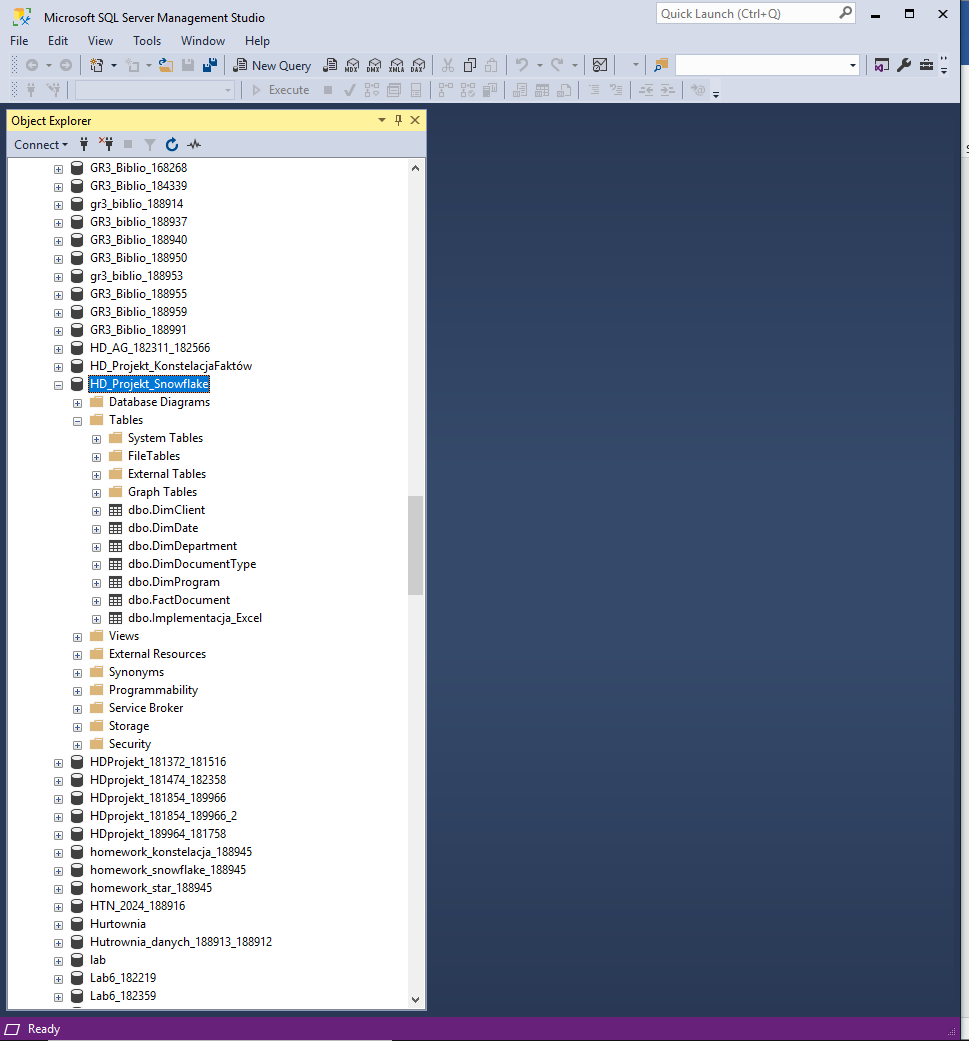


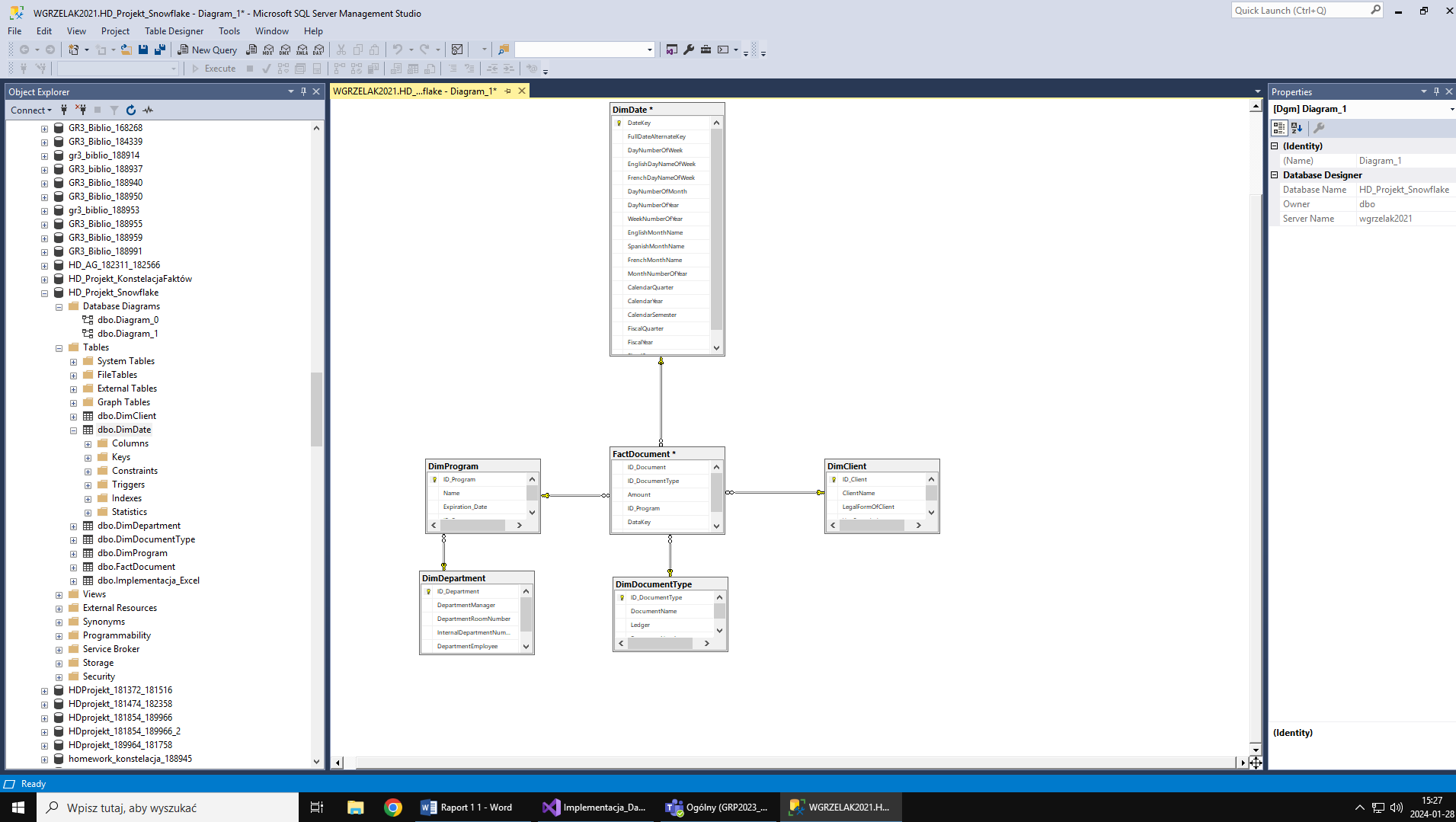
* Ostatnim krokiem jest implementacja. Program nie sygnalizuje błędów, więc tabela powinna zostać poprawnie zagnieźdźona w BD. Implementacja przebiegła pomyślnie.



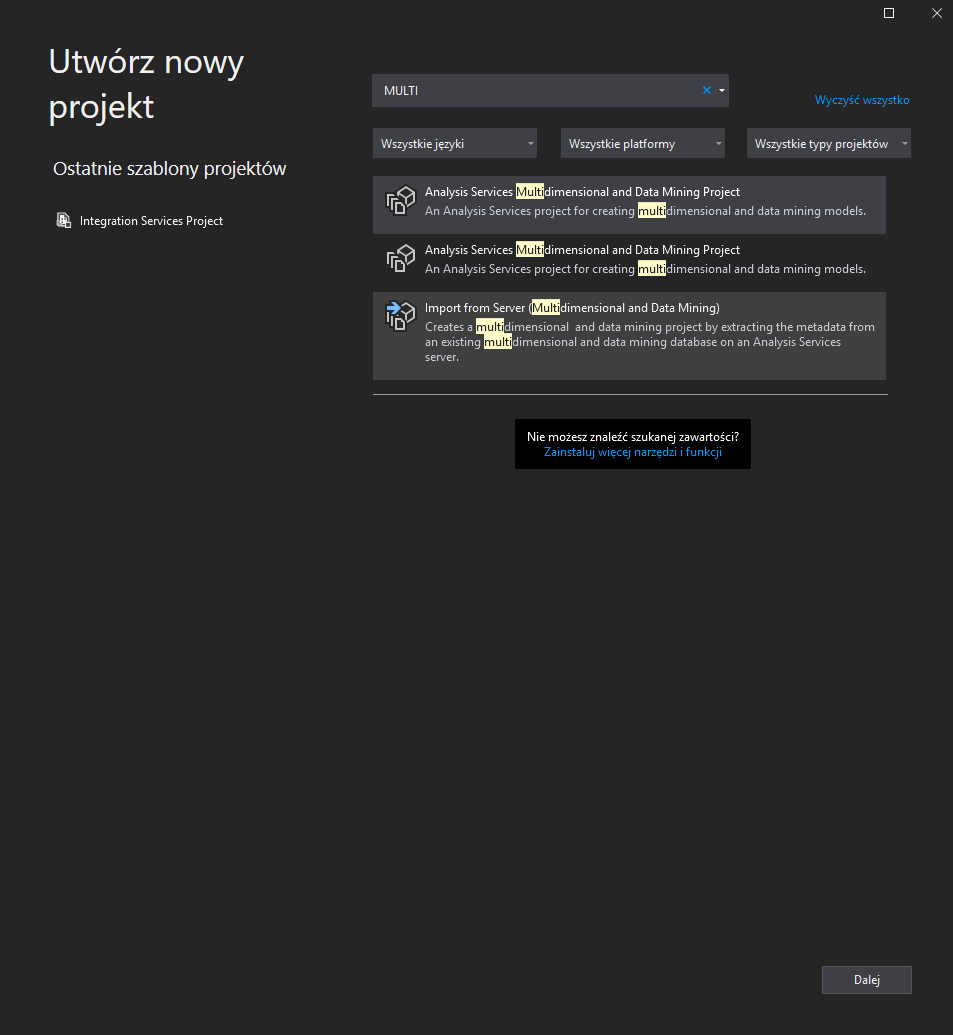
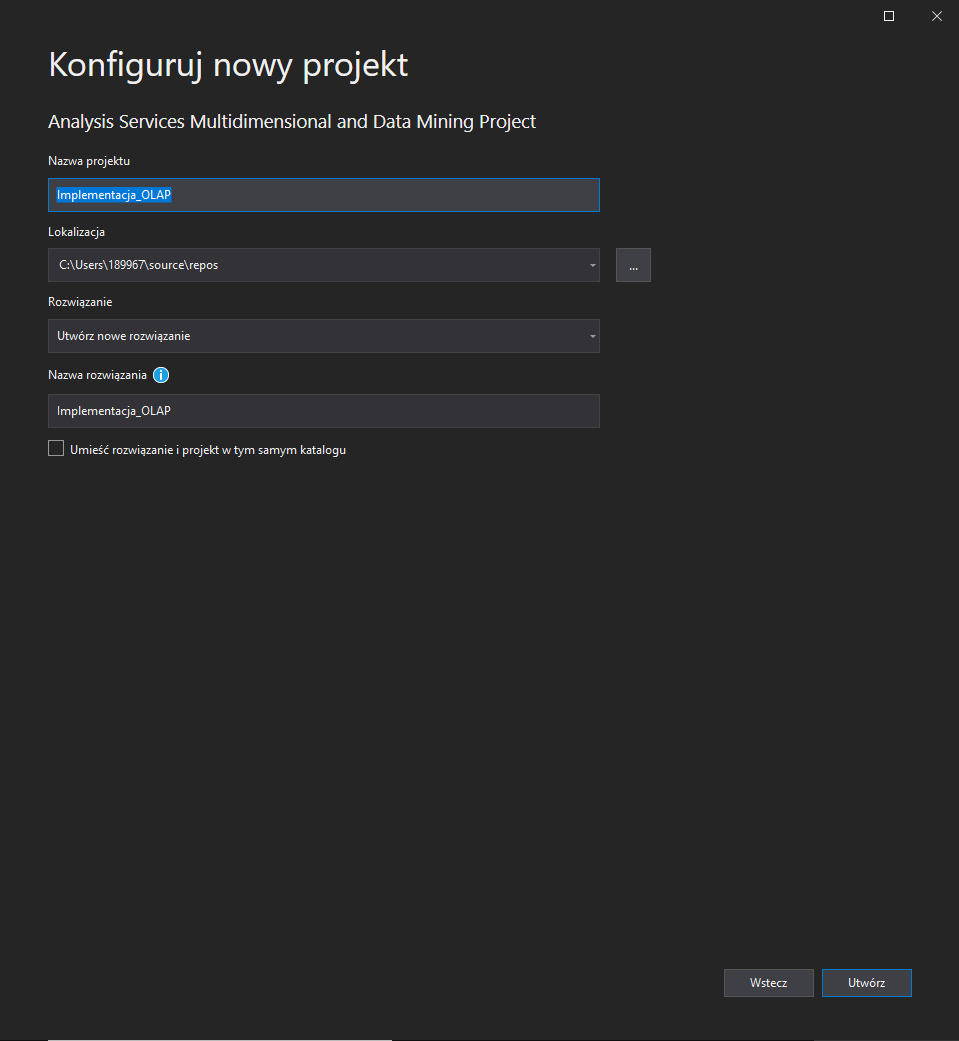
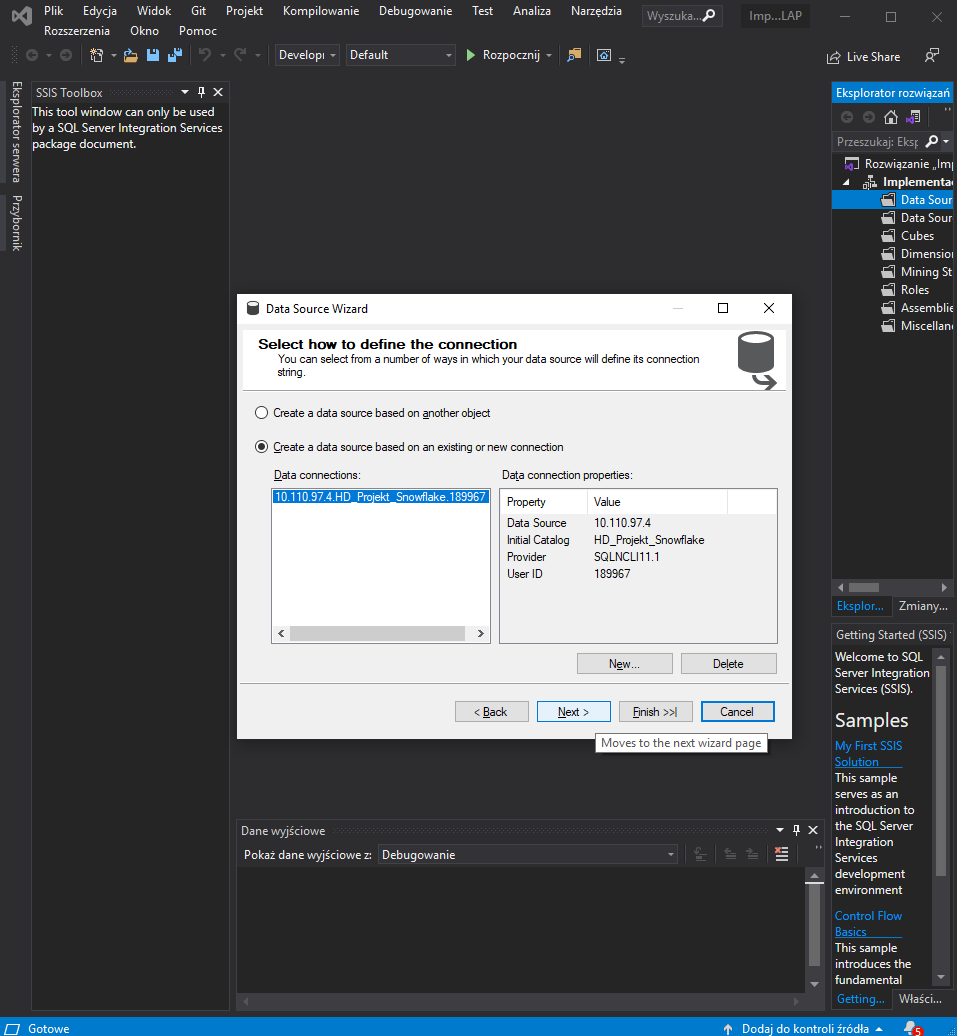
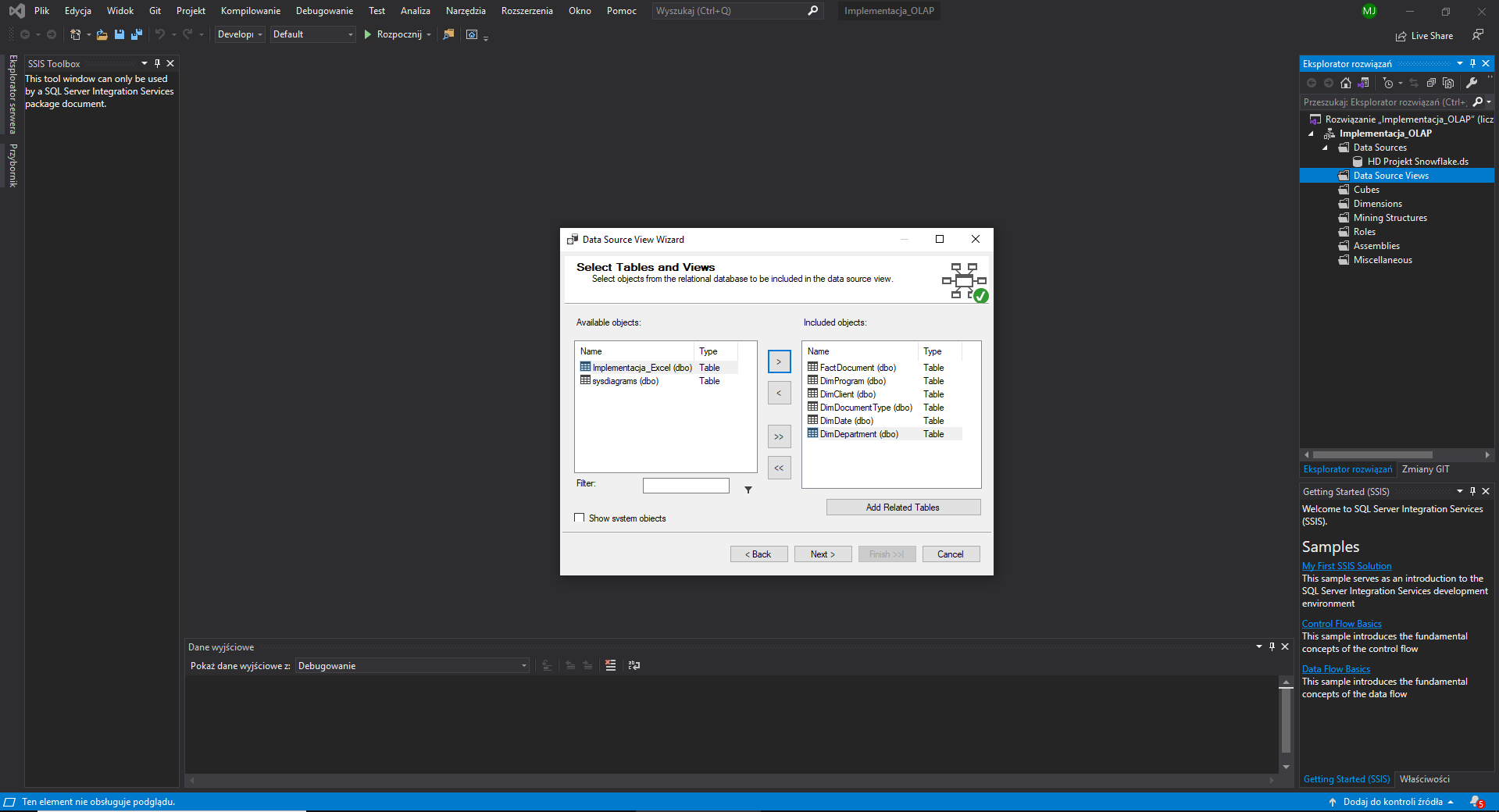
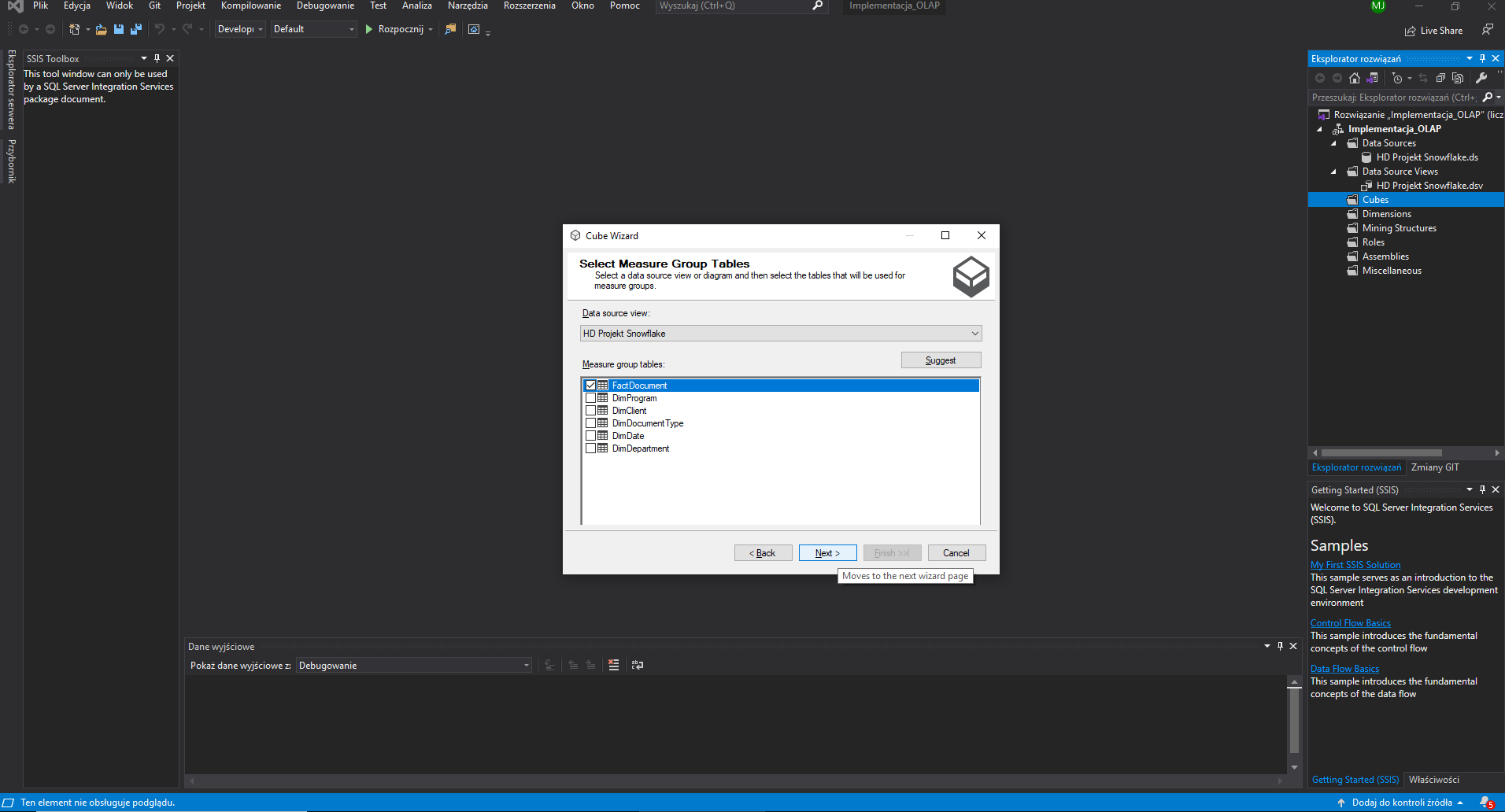
* W kolejnym kroku zostanie wykonana implementacja z wcześniej stworzonego pliku FlatFile zawierającego tabęle DimDate do bazy danych. Ponownie zostaje utworzony Data Flow Task o nazwie Implementacja\_Danych\_FlatFIle. Dodane zostają źródło i miejsce docelowe:

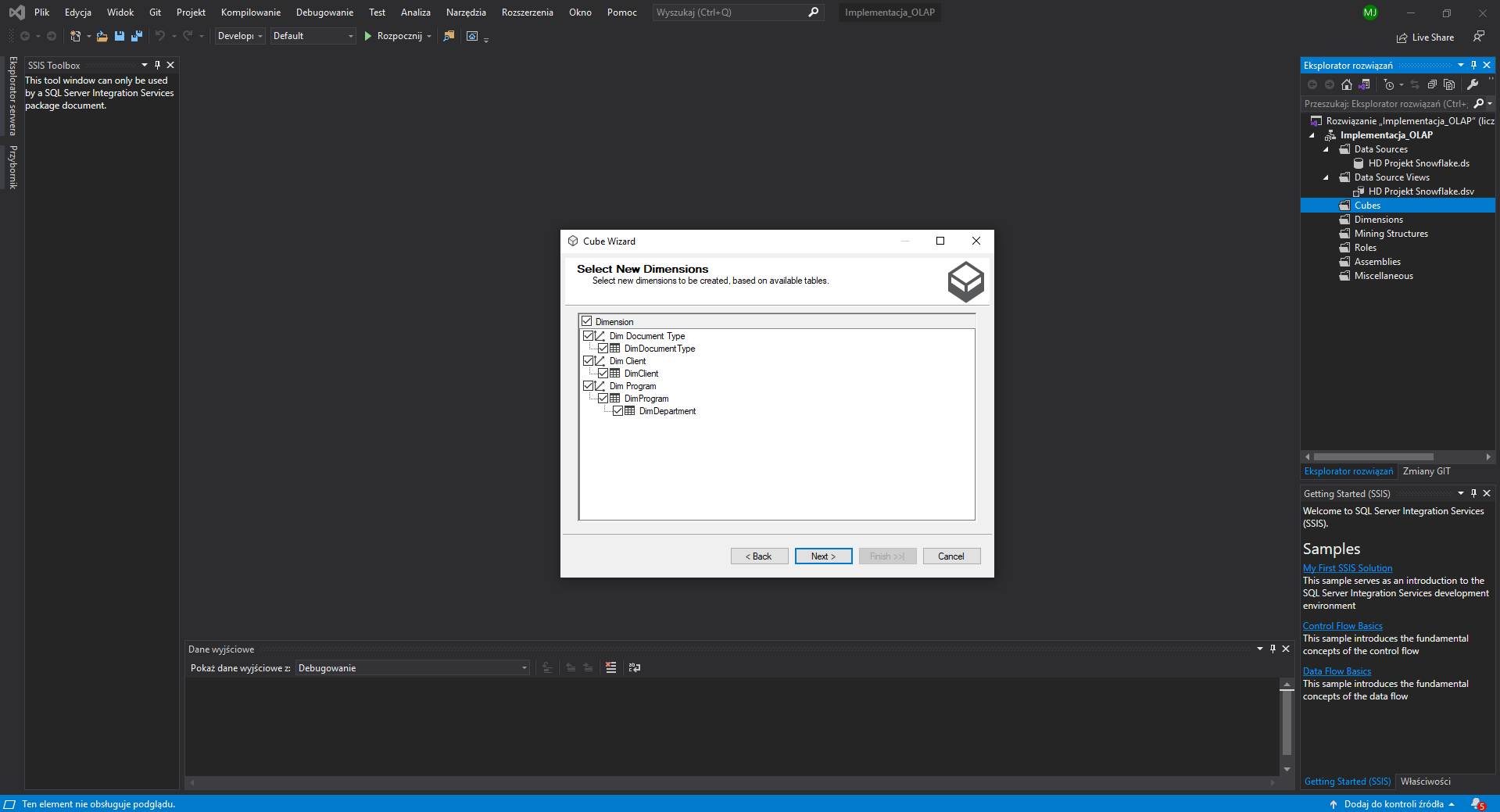


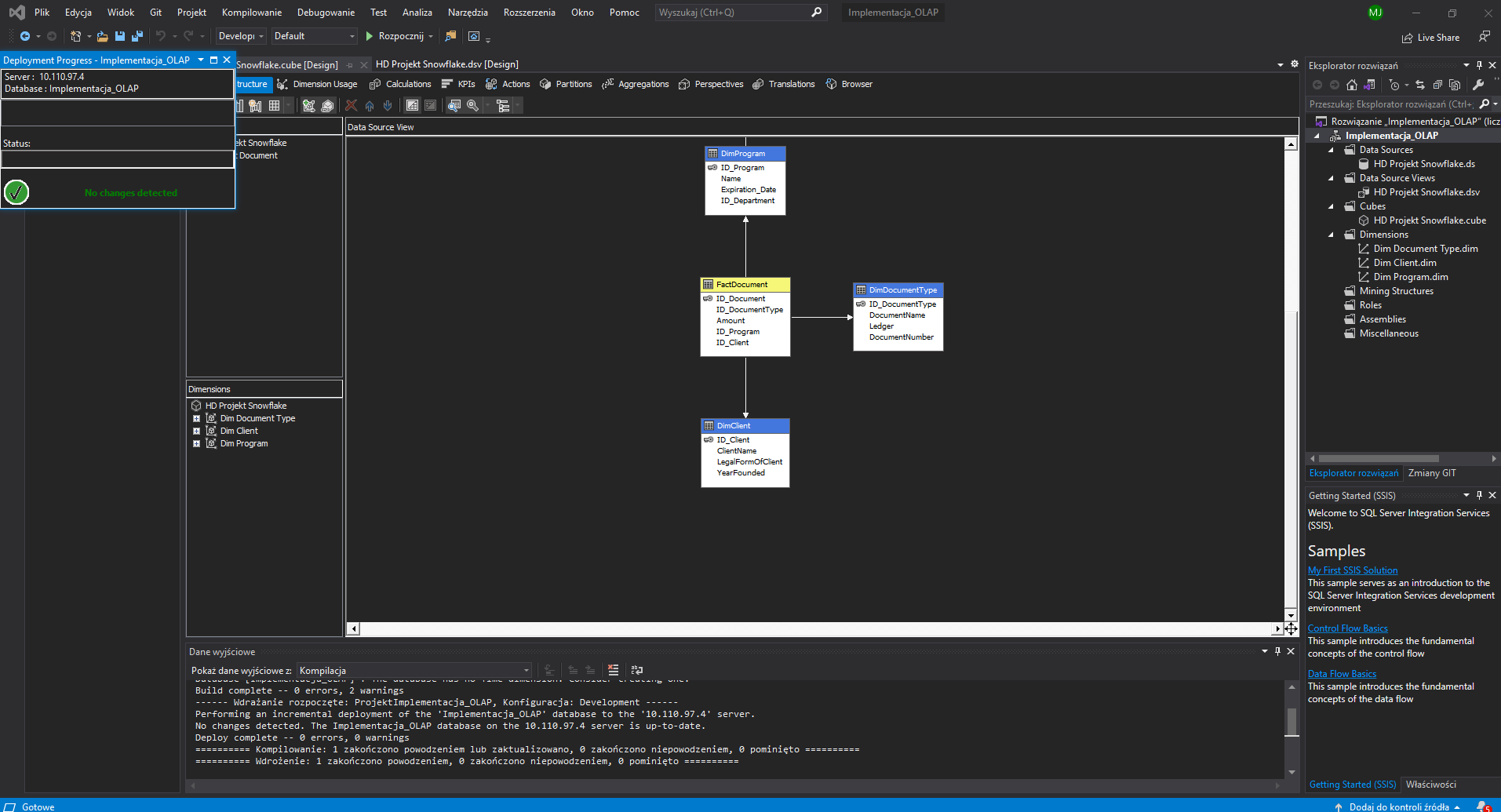
* Dodane zostają połączenia z plikiem typu FlatFile oraz serwerem:
* Przeprowadzona zostaje implementacja i sprawdzenie czy tabela poprawnie została dodana do BD:

Implementacje dla wszystkich trzech rodzajów źródeł zostały wykonane z powodzeniem. Dodatkowo dzięki implementacji udało się dodać brakującą w diagramie tabelę wymiarów dotyczącą Daty (DimDate). Poniżej zaktualizowany diagram:

1. **Implementacja oraz przykładowy przegląd kostki OLAP**

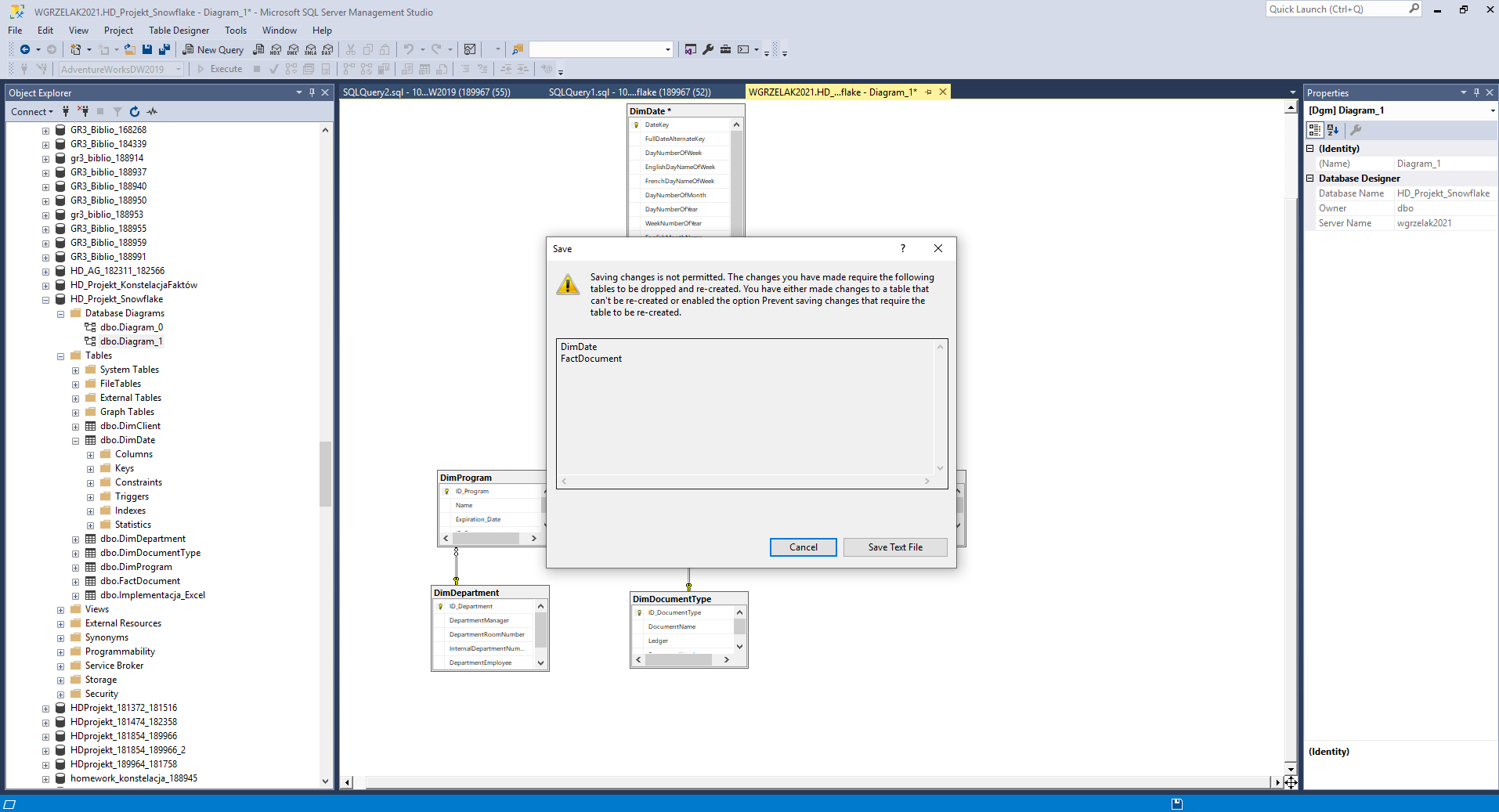
* ****Utworzony zostaje nowy projekt z wykorzystaniem odpowiedniego szablonu. Projekt nazwano Implementacja\_OLAP:
* Zostaje dodane źródło danych, którym będzie wcześniej utworzona Hurtownia z serwera SQL Management Studio:
* Zostaje utworzony nowy widok dla danych, w którym wybrane zostały wykorzystywane w diagramie tabele:
* W kolejnym korku utworzono kostkę OLAP. Wybrana zostaje tabela zawierająca miary tj. FactDocument dla której wybrana została miara Amount czyli kwota dokumentu. Na końcu wytypowano tabelę wymiarów:



* W następnym kroku zmieniona zostaje nazwa serwera na 10.110.97.4 a następnie zostaje wdrożone rozwiązanie:
* Ostatnim krokiem jest pokazanie przykładowego przeglądu kostki OLAP:

Przykładowo dodane rekordy nie pozwalają na przeprowadzenie bardziej zaawansowych modyfikacji kostki w związku czym nie można w przedstawianym przykładzie omówić bardziej funkcjonalności kostki. Jest to wyłącznie kwestia rekordów które gdyby oddawały bardziej realny stan rzeczy mogłyby posłużyć głębszemu omówieniu i pokazanie możliwości zaawansowanego narzędzie jakim jest kostka OLAP.

**Problemy pojawiające się przy realizacji projektu**

Jedynym problemem, który wpłynął na ostateczny kształt projektu było dołączenie tabeli wymiaru DimDate do diagramu. Było to niemożliwe bez usunięcia i odtworzenia tabeli FactDocument oraz dodanie rekordów dla klucza obcego które pokrywałyby się z kolumną DataKey w tabeli DimDate która pełniła rolę klucza głównego. Poniżej komunikat pojawiający się po próbie zaktualizowania diagramu w SQL Management Studio: