Hurtownie danych i systemy Business Intelligence

Dokumentacja projektu

Analiza wyścigów Formuły 1

Autorzy ...
Patryk Świątek
Prowadzący mgr inż. Jakub Abelski

26 listopada 2022

Spis treści

1 Wstęp i cel projektu

Głównym celem podczas projektu jest praktyczne zapoznanie się z tworzeniem architektury oraz modelu składowania i przetwarzania danych w hurtowni danych oraz tworzenia raportów używając systemów Business Intelligence tworząc odpowiednie struktury używając przy tym wybranego przez nas zbioru danych dotyczącego rezultatów wyścigów Formuły 1.

Potencjalny odbiorca rozwiązania będzie miał możliwość analizy wyścigów i tworzenia raportów pod kątem wielu wymiarów takich jak np. kierowcy, konstruktorzy czy warunki pogodowe.

2 Podział ról

W trakcie realizacji projektu role zostały podzielone w sposób następujący:

...:

- Opracowanie schematu architektury działania.
- Stworzenie kostki danych.
- Utworzenie raportów.
- Przeprowadzenie testów potwierdzających poprawność wyników zawartych w raportach.

Patryk Świątek:

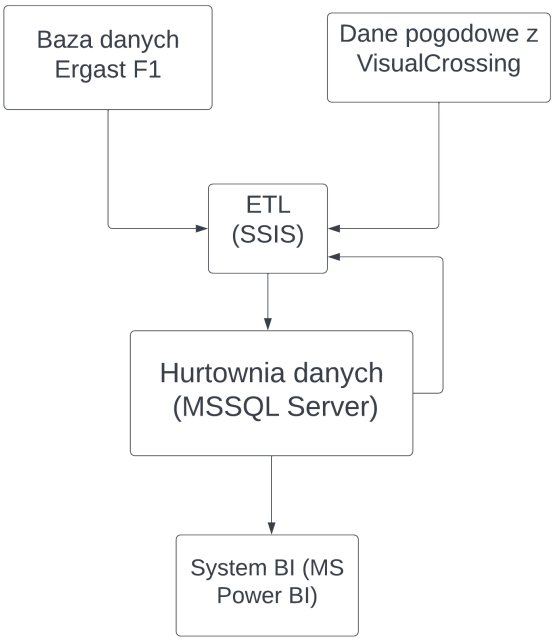
- Opracowanie modelu hurtowni.
- Stworzenie i implementacja procesu ETL dla faktów i wymiarów.
- Przeprowadzenie testów potwierdzających poprawność działania procesu ETL.
- Utworzenie ostatniego wykresu (typu *heatmap*)

Wszystkie procesy i autorskie pomysły przy ich tworzeniu były na bieżąco konsultowane przez wszystkich członków zespołu.

3 Proponowana architektura

W zależności od etapu projektu, efekty będą osiągane przy użyciu architektury dostosowanej do potrzeb danego zadania:

1. Przygotowanie danych źródłowych - Dane ErgastF1 pobrane w postaci relacyjnej bazy danych *Microsoft SQL Server Management Studio*, z kolei dane pogodowe zostały pobrane z API https://www.visualcrossing.com/ z użyciem języka *Python*.
2. Projektowanie modelu hurtowni danych - *Microsoft SQL Server Management Studio*
3. Proces ETL - SSIS (*SQL Server Integration Services*)
4. Załadowanie hurtowni do systemu Business Intelligence i przygotowanie raportów - *Microsoft Power BI*



4 Opis zbiorów danych

4.1 Zbiory danych

Do stworzenia hurtowni użyto bazy danych Ergast zawierającej wyniki kwalifikacji i wyścigów Formuły 1 od 1950 roku. Dostęp do pełnej bazy jest dostępny publicznie. W celu uproszczenia modelu wykorzystano jedynie kilka z wszystkich dostępnych 14 tabel. Na stronie [] jest dostępny gotowy plik .sql, z którego tworzona jest baza danych.

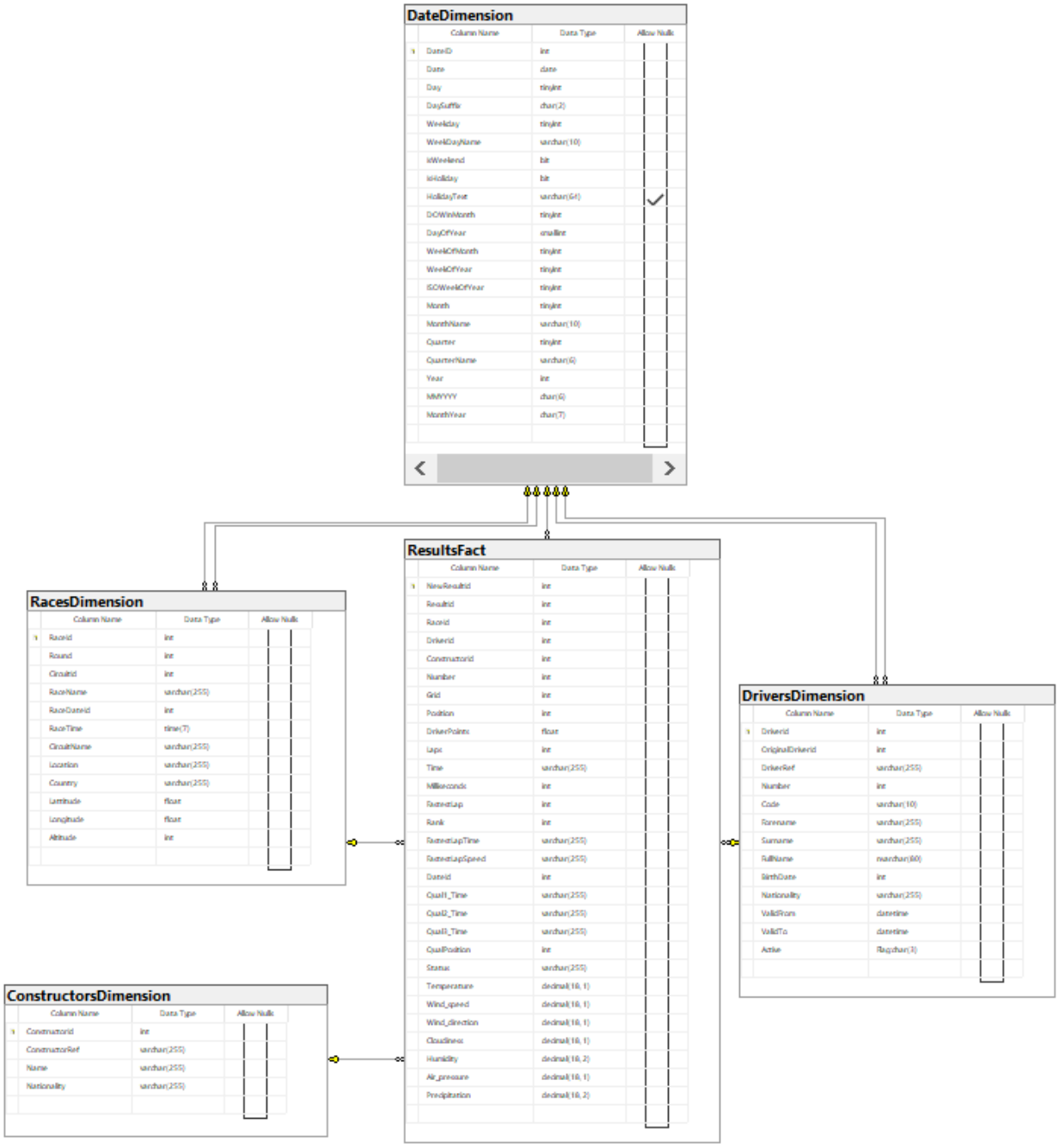
Jako dodatkowe źródło danych zostały użyte dane o warunkach pogodowych w dniu wyścigu. Skorzystano z darmowego API OpenWeather pozwalającego na 1000 żądań dziennie. W związku z dużym zakresem czasu nie jest możliwe uzyskanie danych dla wszystkich możliwych dat. Tabela zawiera podstawowe informacje takie jak informacje o temperaturze, wietrze, wilgotności czy opadach.

4.2 Odświeżanie danych

W związku z dosyć rzadką częstotliwością aktualizacji danych planowane jest sztuczne odświeżanie danych traktując jako startową bazę rezultaty do roku 2021 włącznie, a następnie dodawać kolejne rezultaty wyścigów z obecnego roku w kilkudniowych odstępach.

5 Model hurtowni danych

Poniższy diagram przedstawia zaproponowany model hurtowni danych, utworzony w programie *Microsoft SQL Server Management Studio*.



Zaproponowany został model gwiazdy, której fakt stanowi wynik osiągnięty w danym wyścigu. Utworzone zostały następujące tabele wymiarów:

- wymiar daty
- wymiar wyścigu - przedstawia atrybuty związane z pojedynczym wyścigiem

- wymiar konstruktorów
- wymiar kierowców

Dodatkowo w tabelach wymiarowych zdefiniowane zostały zmienne odnoszące się do wymiaru daty tzw. *outrigger*, na przykład zmienna *BirthDate* w tabeli *DriversDimension*

W modelu wymiar *DriversDimension* został zdefiniowany jako *Slowly changing dimension*. W założeniu każda zmiana w wymiarze zostanie zanotowana metodą śledzenia zmian (*track changes*).

Do celów raportowych zostanie też stworzona hierarchia: *Country* - *Location* - zmienne w tabeli *RacesDimension*.

6 Kluczowe miary i atrybuty

6.1 Kluczowe miary

W tabeli faktów omawianego modelu zostały zdefiniowane następujące kluczowe miary:

- Pozycja końcowa (*Position*) - zmienna całkowitoliczbowa. W bazie źródłowej dopuszczalne są braki danych ze względu na możliwy brak ukończenia lub klasyfikacji danego zawodnika w wyścigu - w hurtowni zamienione na wartości równe zero.
- Osiągnięty czas przejazdu (*Milliseconds*) - zmienne odpowiednio typu *time* oraz całkowitoliczbowa. Miara kluczowa do porównania czasów osiągniętych w wyścigu przez poszczególnych zawodników. W bazie źródłowej danych dopuszczalne braki danych ze względu na możliwość nieukończenia wyścigu przez kierowcę - w modelu hurtowni zostaną one zamienione na wartości tekstowe czytelniejsze w raportach dla użytkownika.
- Punkty zdobyte przez kierowcę w wyścigu (*DriverPoints*) - miara bardzo istotna w kontekście analizy wyników zawodników na przestrzeni dłuższego okresu np. w całym sezonie.
- Pozycja startowa (*Grid*) - miara niezbędna do analizy zależności zajętego miejsca na koniec wyścigu od miejsca zajmowanego na początku. Jest to zmienna całkowitoliczbowa i z oczywistych względów nie są dopuszczane braki wartości.
- Ilość opadów czy temperatura w dniu danego wyścigu (*Precipitation*, *Temperature*) - ważne miary w kontekście analizy zależności występów zawodników, a szczególnie ich czasów, od warunków pogodowych (są to zmienne całkowitoliczbowe).

Poza wyżej wymienionymi miarami, ważne w kontekście raportowania mogą być również np. miejsce zajęte w kwalifikacjach do wyścigu czy czas najszybciej przejechanego okrążenia przez danego zawodnika.

6.2 Kluczowe atrybuty

Spośród wszystkich atrybutów występujących w proponowanym modelu najbardziej kluczowe są:

- imię i nazwisko kierowcy (*Forename, Surname* w tabeli *DriversDimension*) - niezbędne atrybuty do analizy i raportowania rezultatów poszczególnych zawodników. Oba atrybuty są oczywiście zmiennymi tekstowymi.
- nazwa toru, oficjalna nazwa wyścigu (odpowiednio *CircuitName* i *Name* w tabeli *RacesDimension*) - zmienne tekstowe, bardzo istotne przy analizie i porównaniu rezultatów w danym wyścigu czy wyścigach na danym torze na przestrzeni kilku lat.

- Nazwa marki bolidów (*Name* z tabeli *ConstructorsDimension*) - konieczna przy analizie osiąganych wyników bolidów danej marki na tle innych (zmienna tekstowa bez żadnych braków wartości).

7 Opis procesu ETL

Proces ETL został wykonany w narzędziu SSIS (*SQL Server Integration Services*) w ramach dwóch pakietów SSIS:

- *LoadDimensions* - zawiera poszczególne etapy procesu ETL zdefiniowane w celu zasilenia tabel wymiarowych odpowiednio przetworzonymi danymi
- *LoadFacts* - pakiet stworzony do zdefiniowania procesu ETL dla tabeli faktowej.

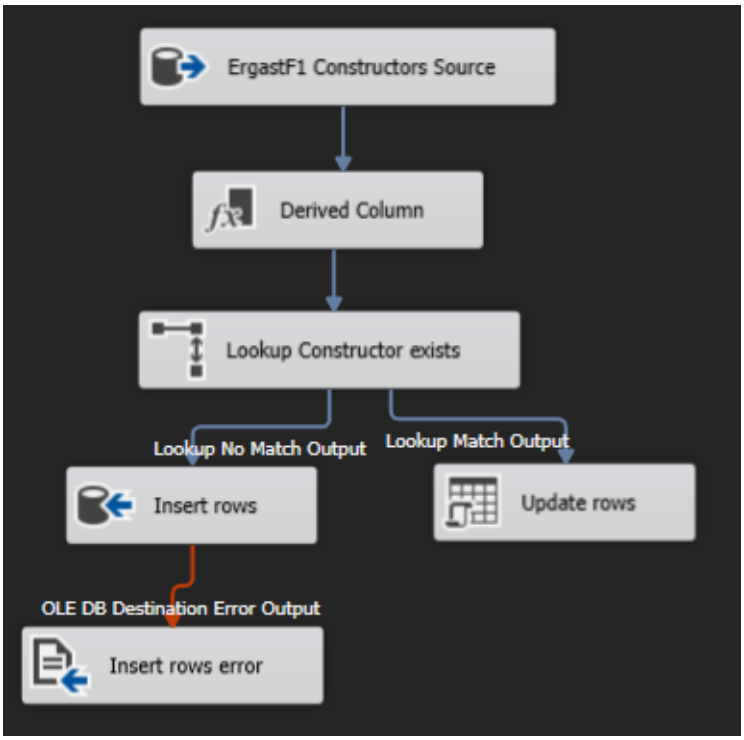
Poszczególne części całego procesu ETL zostaną szczegółowo omówione w poniższych podsekcjach.

7.1 ETL dla tabel wymiarowych

W ramach tej części dla każdego wymiaru (oprócz wcześniej wygenerowanego wymiaru daty) zostały zdefiniowane tzw. *Data Flow tasks* umożliwiające przepływ danych z tabel źródłowych bazy transakcyjnej do odpowiedniej tabeli wymiarowej.

7.1.1 Wymiar konstruktorów

Poniżej przedstawiony został schemat przepływu danych dla tej części:

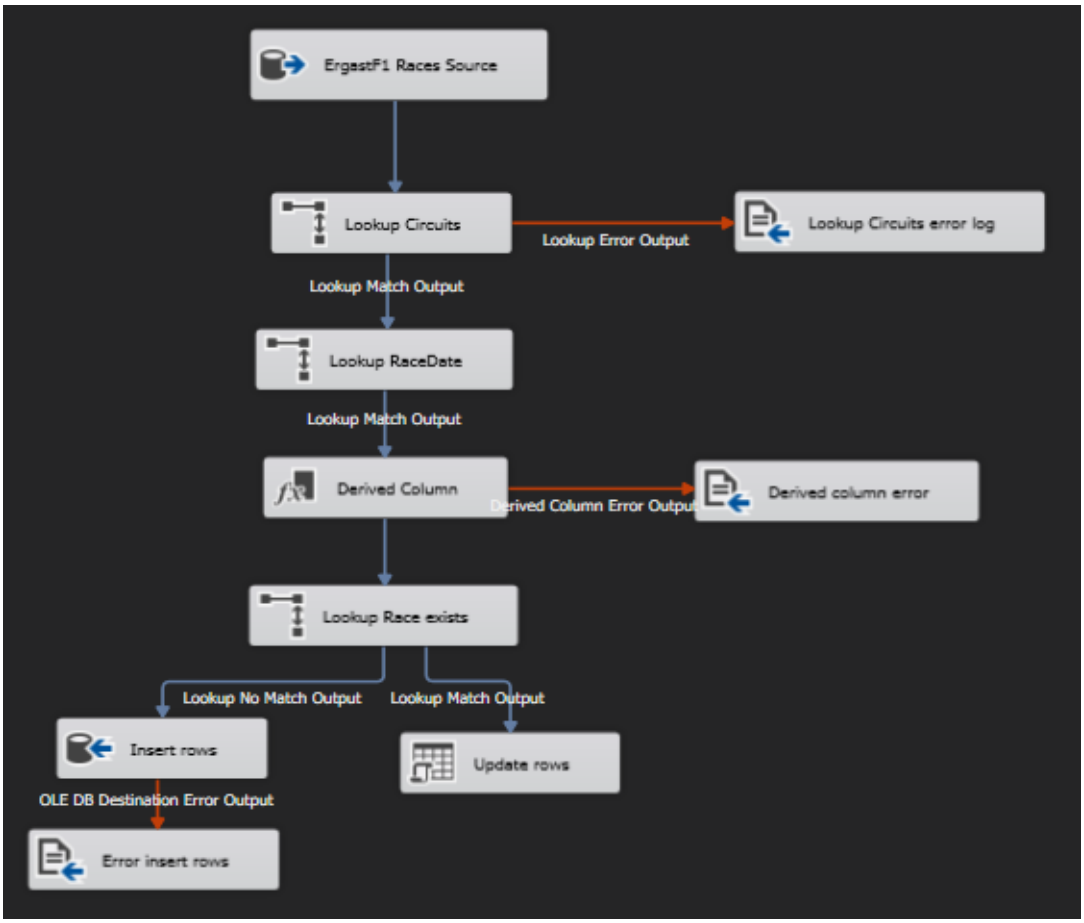


Składa się on z:

- Załadowania danych z tabeli *constructors* z bazy *ErgastF1*.
- Przekształcenia kolumny *Nationality* polegającego na zamianie brakujących danych na wartości "UNKNOWN".
- Sprawdzeniu czy Konstruktor o danym ID już istnieje w hurtowni. Jeśli tak, to aktualizujemy jego atrybuty. W przeciwnym wypadku ładujemy rekordy do tabeli wymiarowej. Dodatkowo w celu sprawdzenia ewentualnych błędów ładowania stworzony został plik, do którego wpływają rekordy, których ładowanie do hurtowni zakończone zostało błędem.

7.1.2 Wymiar wyścigów

Schemat procesu:



Główną tabelą źródłową dla tego wymiaru jest tabela *races*. Za pomocą procedury *lookup* zostaje do niej dołączona tabela *circuits*, która daje źródło atrybutom informującym o najważniejszych cechach torów wyścigowych.

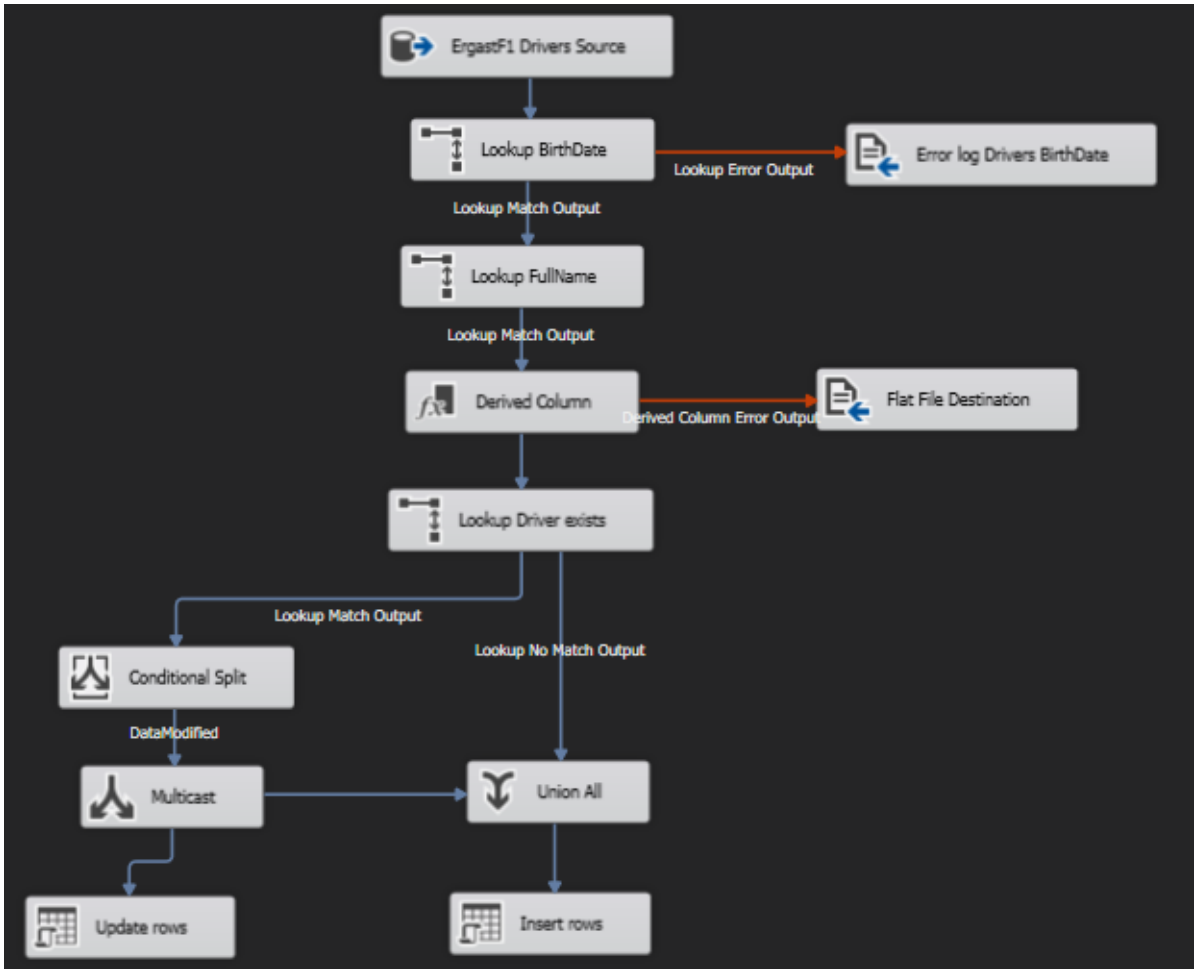
Następnie tworzony jest *outrigger* odwołujący się do wymiaru daty i informujący o dacie wyścigu.

Kolejne etapy obejmują:

- Przetwarzanie zmiennych poprzez zastępowanie brakujących wartości wartościami odpowiednimi dla danego typu (dla zmiennych tekstowych *Country*, *Location* - "UNKNOWN", dla zmiennej inofrmującej o wysokości danego toru będzie to wartość liczbowa 10000, z kolei dla zmiennej czas - "00:00:00.0000000")
- Sprawdzenie czy rekord o danym *raceId* już istnieje - jeśli tak, to aktualizujemy jego niektóre atrybuty, jeśli nie - wstawiamy nowy rekord do tabeli.

7.1.3 Wymiar kierowców

Schemat procesu:



Przepływ danych jest dosyć podobny do tych z wymiaru wyścigów czy konstruktorów. Również tworzony jest *outrigger* odwołujący się do wymiaru daty (data urodzin kierowcy). W ramach przetwarzania danych usuwane są w odpowiedni dla typu danych sposób brakujące wartości dla zmiennych *Number, Code*.

Dodatkowo za pomocą *Lookup* tworzona jest nowa zmienna *FullName*, która łączy imiona i nazwiska poszczególnych zawodników.

Jednak główną różnicą dla tego wymiaru jest obsługa przepływu danych zgodnie z techniką dla typu *slowly changing dimension*. W trakcie procesu ETL stworzona została zmienna *LoadTime* przechowująca datę ładowania danych. Podobnie jak w poprzednich przypadkach zdefiniowany został etap sprawdzenia, czy rekordy o danym ID już istnieją w hurtowni. Jednakże w przypadku, gdy taki rekord istnieje nie jest on aktualizowany, ale (po sprawdzeniu czy któraś zmienna uległa zmianie w ramach *Conditional Split*) wstawiony do tabeli z nowym ID, a oryginalne ID zapisywane jest w kolumnie *OriginalDriverId* oraz zmienna *ValidFrom* przyjmuje wartość zmiennej *LoadTime*. Taką wartość też w ramach aktualizacji rekordów istniejących o danym ID przyjmuje zmienna *ValidTo*.

7.2 ETL dla tabeli faktów

Proces zasilający dane faktowe dzieli się na trzy główne etapy:

- Ucięcie z tabeli rekordów już istniejących.
- Stworzenie tzw *staging table*, która przechowuje wszystkie rekordy napływające z tabel źródłowych.
- Zdefiniowanie i zasilenie tabeli faktowej danymi przy zachowaniu odpowiednich relacji.

7.2.1 Staging Table

Schemat:

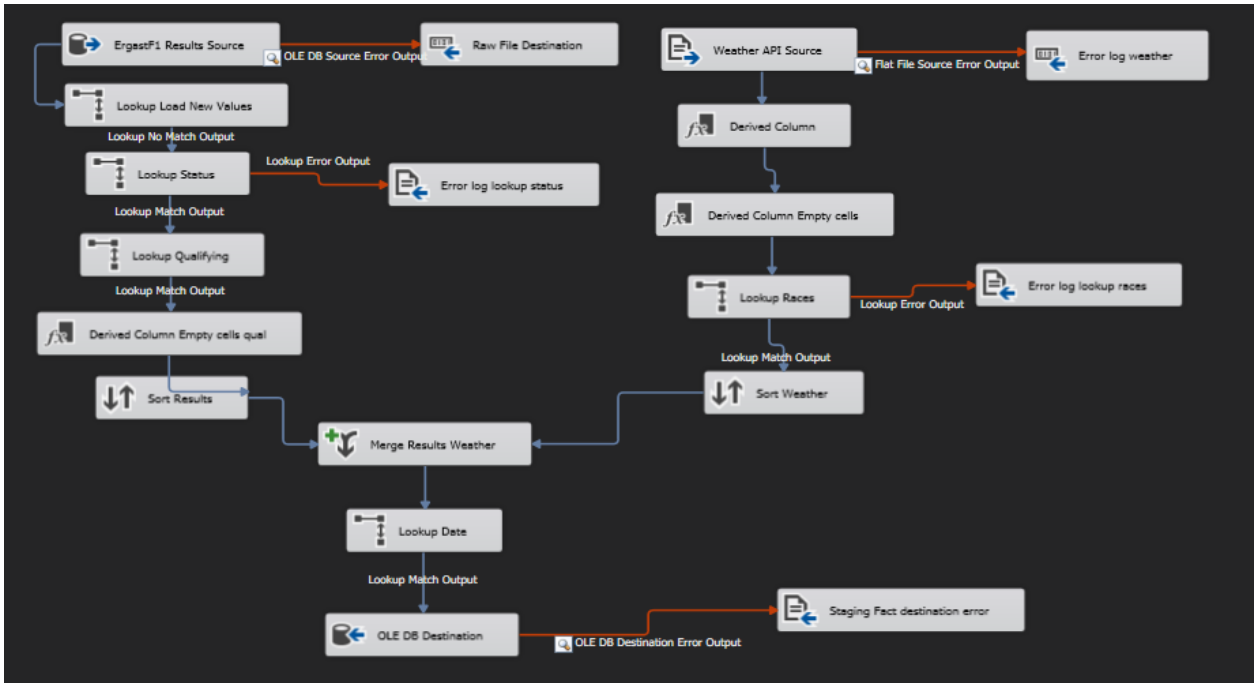


Tabela jest generowana z dwóch źródeł:

- Baza ErgaztF1 - głównie tabela *results*, oprocz tego zdegenerowana do faktów została tabela *status* (informująca o statusie ukończenia wyścigu) oraz tabela *qualifying* (a konkretnie atrybuty pozycji po kwalifikacjach oraz poszczególnych czasów kwalifikacji).

- plik .txt z tabelą pogodową wygenerowaną poprzez zapytania API. Po załadowaniu zmienne docelowo numeryczne (po wygenerowaniu były to zmienne przypisane jako tekstowe) zostały w następujący sposób przetworzone:
  - usunięte zostały niepotrzebne spacje,
  - puste wartości poszczególnych zmiennych zostały zamienione na braki danych.

  Następnie w ramach *Lookup Races* do tabeli dołączona została zmienna *raceId*. Nastąpiło to poprzez łączenie tabeli pogodowej ze źródłową tabelą wyścigów poprzez zmienne daty (przyjęte zostało założenie, że w ciągu jednego dnia odbywa się jeden wyścig formuły 1).

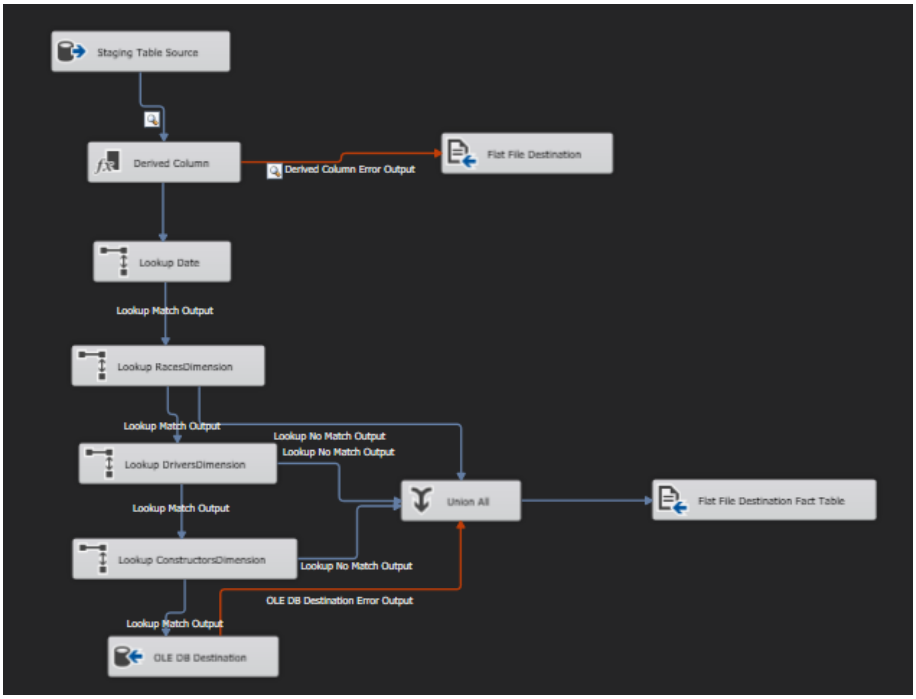
Obie tabele przetworzone w sposób opisany powyżej zostały ze sobą połączone, a dokładnie do tabeli z rezultatami została dołączona tabela pogodowa za pomocą *Left outer join*. Ten sposób łączenia wynika z braku źródłowych danych pogodowych sprzed 1970 roku.

Po złączeniu tabel została utworzona zmienna *RaceDate* zimportowana z tabeli *races* za pomocą zdefiniowanego *Lookup Date*. Ruch ten był konieczny z powodu braków danych zmiennej daty po złączeniu dwóch przetworzonych tabel źródłowych (braki te występowały dla wyników wyścigów odbywających się przed 1970 rokiem).

Ostatecznie wynikowa tabela została załadowana do pomocniczej tabeli w bazie danych.

7.2.2 Proces dla docelowej tabeli faktów

Schemat:



Przyjmując w tej fazie jako źródło danych *staging table*, pozostało niewiele zmian do wprowadzenia w danych. Najważniejszym punktem przetwarzania jest zamiana brakujących wartości, przykładowo:

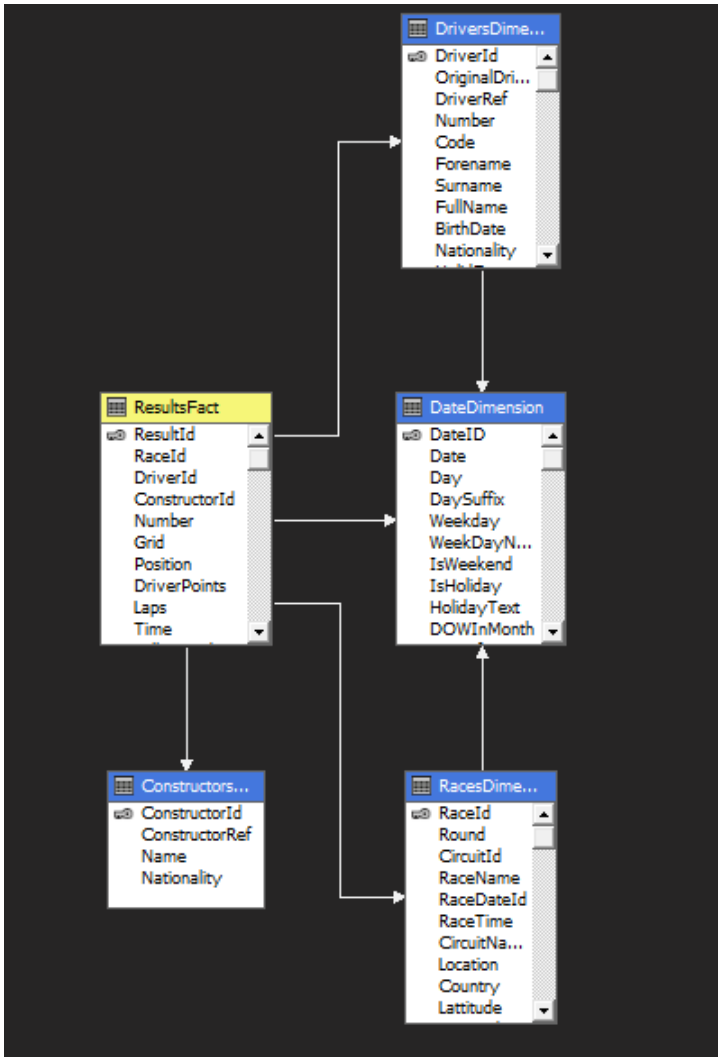
- dla zmiennych pogodowych poza temperaturą są to wartości równe -1, z kolei dla temperatury wartość -100,

- dla zmiennych określających np. końcową pozycję po wyścigu czy kwalifikacjach, numer najszybszego okrążenia to zastępujemy je wartością równą 0
- dla zmiennych tekstowych np. dla czasu i strat czasowych wyścigu - wartość 'NOT MEASURED'.

Następnie za pomocą złączenia z wymiarem daty definiujemy kolumnę z ID daty. Definiujemy też procedury *Lookup* w celu sprawdzenia zgodności zmiennych będących kluczami obcymi z kluczami podstawowymi poszczególnych wymiarów. Ostatecznie rekordy są ładowane do docelowej tabeli faktów w hurtowni.

8 Opis warstwy raportowej

Kostkę danych stworzono w narzędziu SSAS (SQL Server Analysis Services), a same raporty w narzędziu Microsoft Power BI Desktop. Jako źródło kostki danych użyto oczywiście gotowej hurtowni danych po użytym ETL, widok po załadowaniu hurtowni widok źródła wyglądał następująco

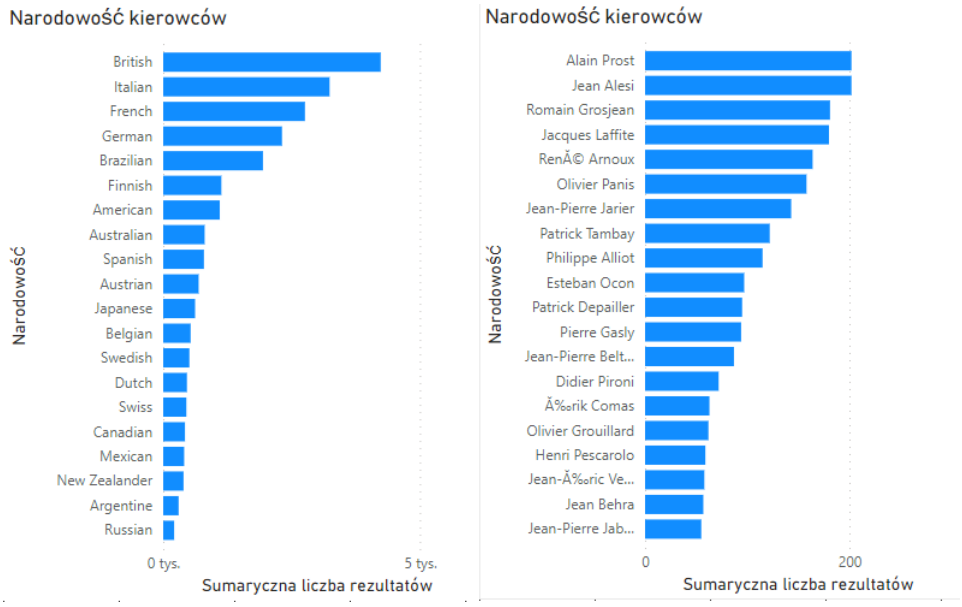


W narzędziu MS Power BI utworzono następujące hierarchie:
Dla wymiaru *RaceDimension*:

- Country
- Circuit Name
- Race Name

Przykładowe użycie hierachi w raporcie *Popularność F1 w różnych krajach*, klikając w *UK* widzimy, że wyścigi odbywały się na czterech różnych torach, potem klikając na *Brands Hatch* widzimy, że odbywało się tam *British Grand Prix* oraz *Europan Grand Prix*.

Dla wymiarów *DriversDimension* oraz *ConstructorsDimension* utworzono hierarchie dla narodowości. Przykładowe użycie w tym samym raporcie - kilkając na French otrzymujemy kierowców z Francji.



Do utworzenia raportu *Wpływ pogody na wyniki* utworzono kolumnę agregującą rezultaty pod wpływem tego czy w danym wyścigu wystąpiły opady.

```
1  Rain = IF('Ergast F1 DWH'[Precipitation] > 0, "Rain", IF('Ergast F1 DWH'[Precipitation] = 0, "No Rain", "No weather data"))
```

W raporcie *Pozycja startowa i pozycja końcowa* stworzono kolumnę grupującą rezultaty po pozycji startowej kierowców.

```
Grid grouped = IF('Ergast F1 DWH'[Grid] <= 5, "1-5", IF('Ergast F1 DWH'[Grid] <= 10, "6-10", IF('Ergast F1 DWH'[Grid] <= 15, "11-15", "16-")))
```

Natomiast w wielu raportach zamiast zwykłej miary użyto miary sprawdzającej czy dany kierowca ukończył tzn. czy miara Position jest dodatnia. Użyto średniej jako agregacji.

```
1  PositionFinished = CALCULATE(AVERAGE('Ergast F1 DWH'[Position]), FILTER('Ergast F1 DWH', 'Ergast F1 DWH'[Position] > 0))
```

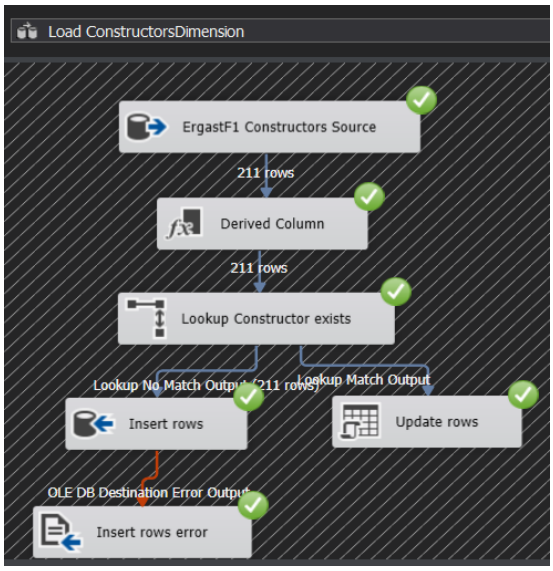
9 Testy

W tej sekcji przedstawione zostaną różne testy sprawdzające poprawność działania procesu ETL, ładowania danych do hurtowni czy raportowania w narzędziu BI.

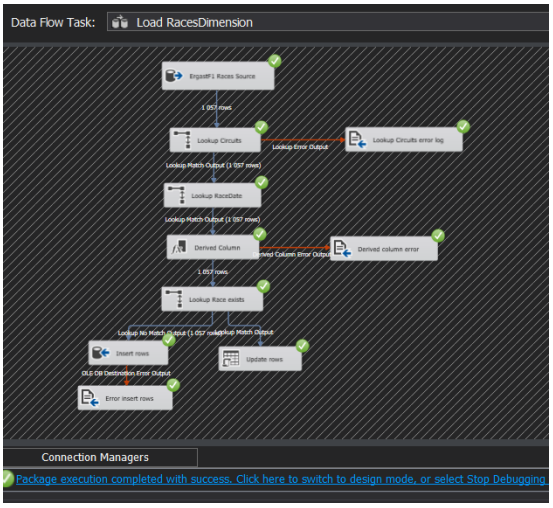
9.1 Potwierdzenie poprawnego działania poszczególnych procesów ETL

Poniżej przedstawione są zrzuty ekranu pokazujące poprawne załadowanie wszystkich zdefiniowanych procesów SSIS w sytuacji gdy jedyna tabela w hurtowni, która jest zasilona danymi to wymiar daty. Podstawowa baza obejmuje dane z wyścigów do końca 2021 roku. Następnie załadujemy dane z roku bieżącego.

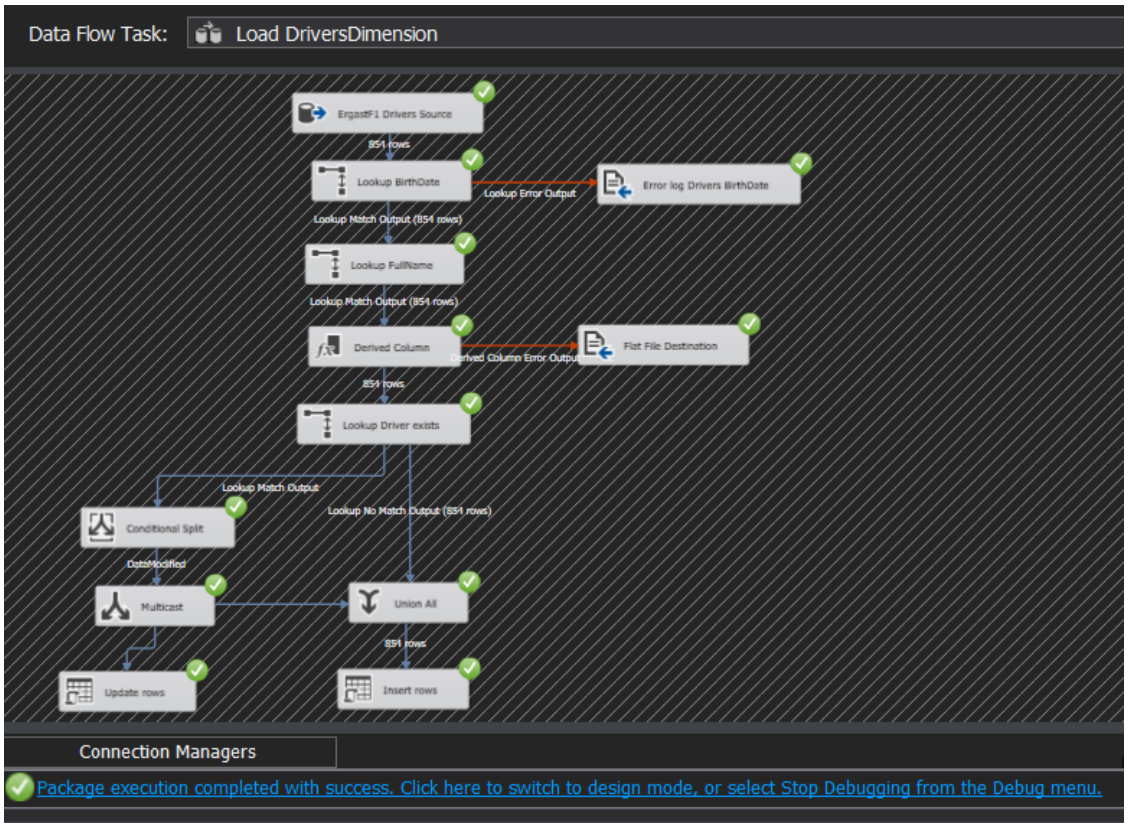
Chcemy porównać później tabele oryginalne z załadowanymi tabelami w hurtowni (szczególnie liczność ich wierszy).

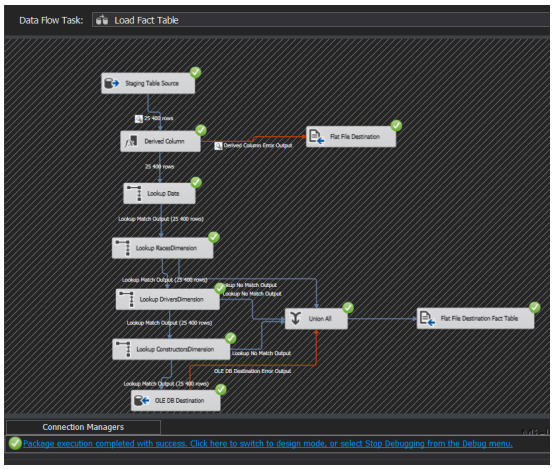


(a) Constructors Dimension

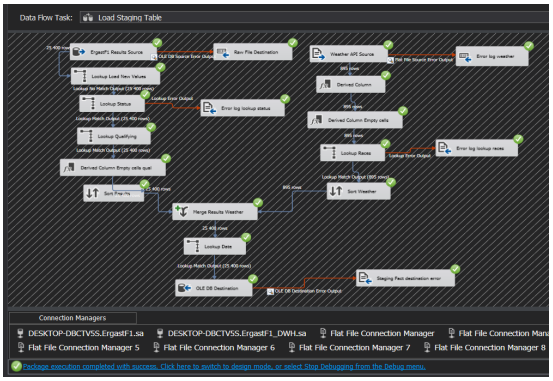


(b) Races Dimension





(a) Tabela faktów



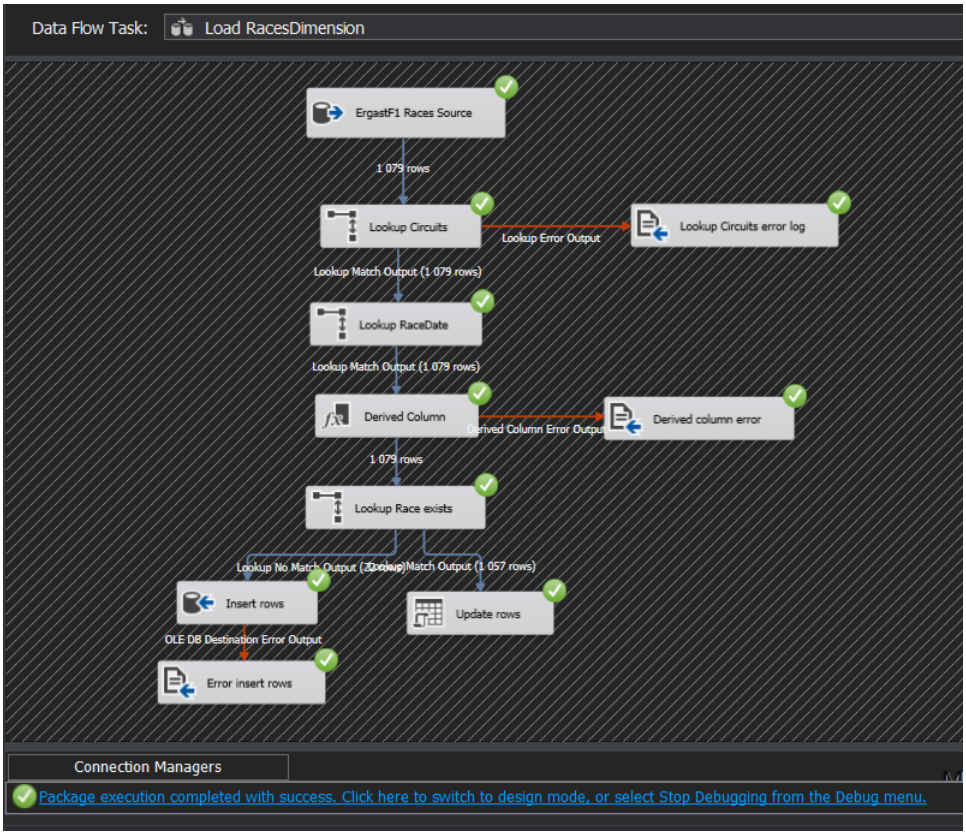
(b) Staging Table

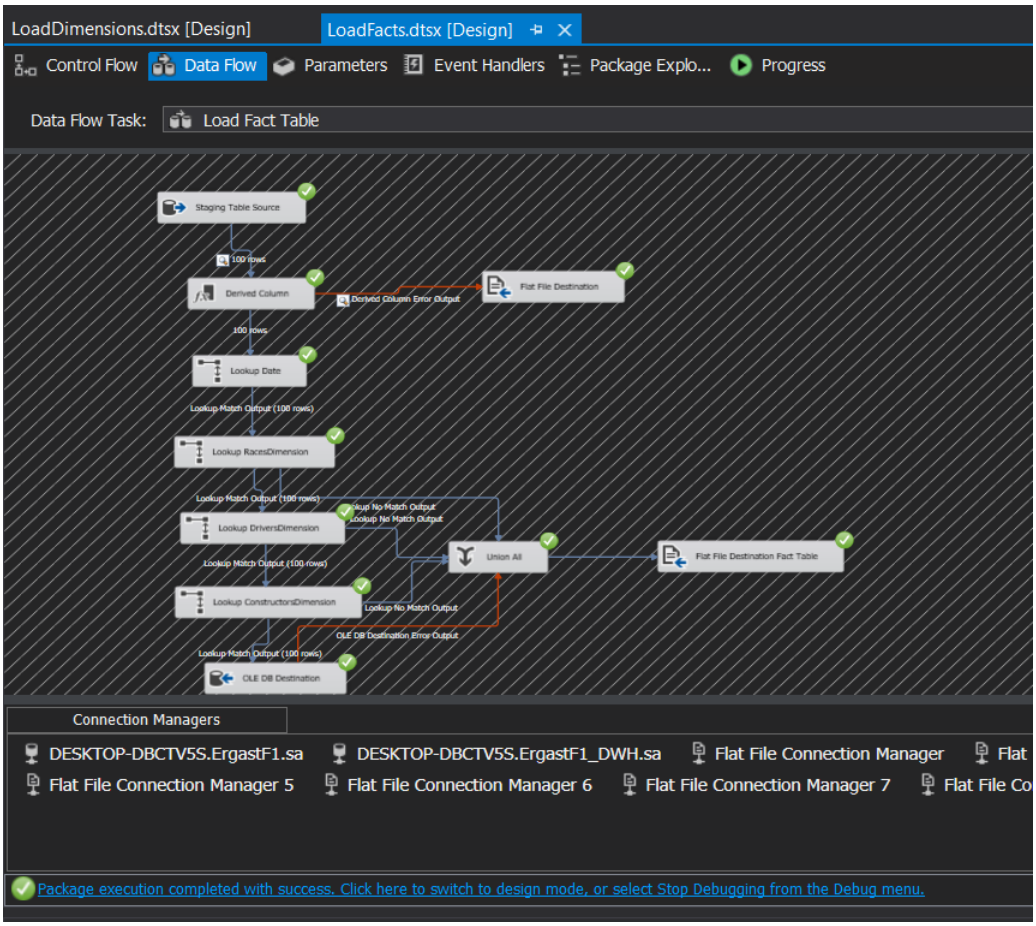
Liczba wierszy załadowanych do następujących tabel:

- konstruktorzy - 211
- kierowcy - 854
- wyścigi - 1057
- wyniki (fakty) - 25400

W celu sprawdzenia poprawności zasilania hurtowni nowymi danymi, ponownie uruchomiona zostanie procedura ETL dla zaktualizowanych table źródłowych o dane z bieżącego roku.

Nowe dane źródłowe pojawiły się w tabelach *races* oraz *results*, zatem spodziewamy się zmian w wymiarze wyścigów oraz w faktach. Proces ponownego przepływu danych wyglądał następująco:





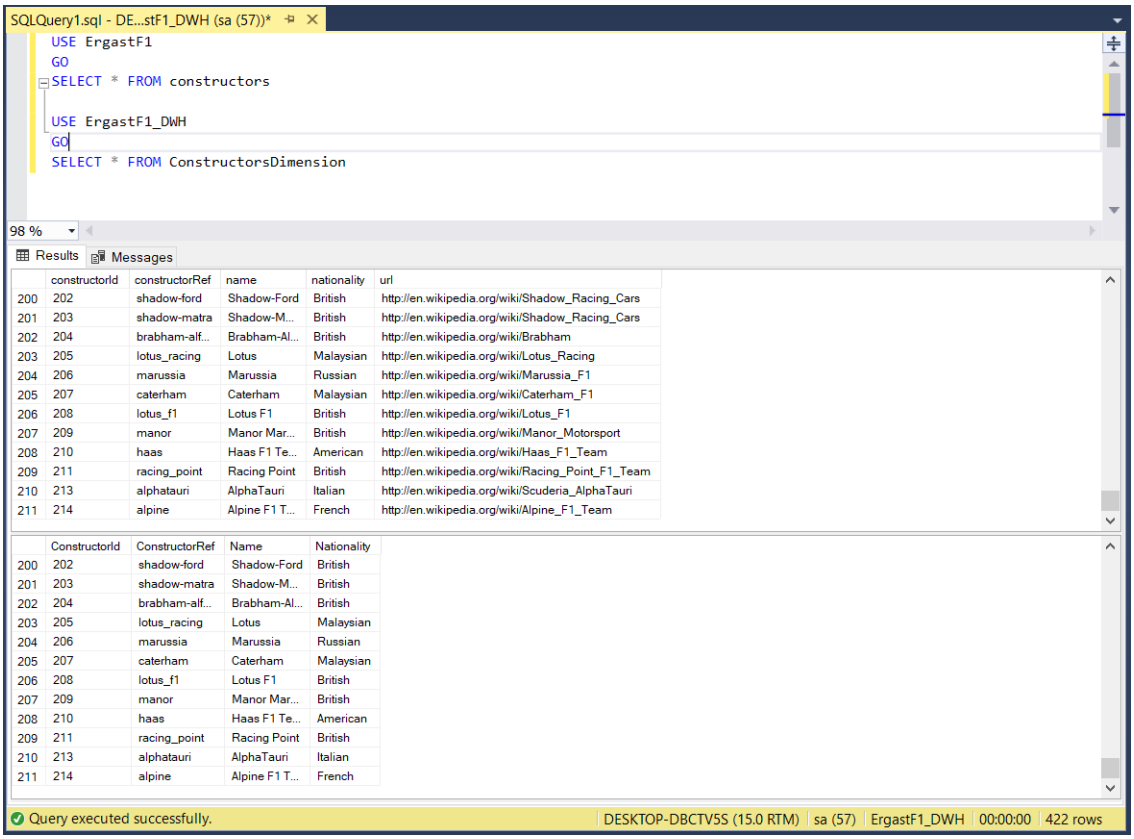
Możemy zauważyć, że ze źródła danych wyścigów zostało pobranych 1079 wierszy, z czego 1057 zostało wykrytych już w hurtowni, a reszta dodana jako nowe rekordy. Zgadza się to z ilością wierszy zasilających ten wymiar podczas inicjalizacji.

Z kolei SSIS dla tabeli faktów załadował do niej 100 dodatkowych wierszy, a proces zakończył się poprawnie. Widzimy zatem, że sam proces ETL zakończył się nie wykazując żadnych błędów. W następnej podsekcji sprawdzimy, czy dane w hurtowni pokrywają się ze źródłowymi.

9.2 Testowanie danych w hurtowni

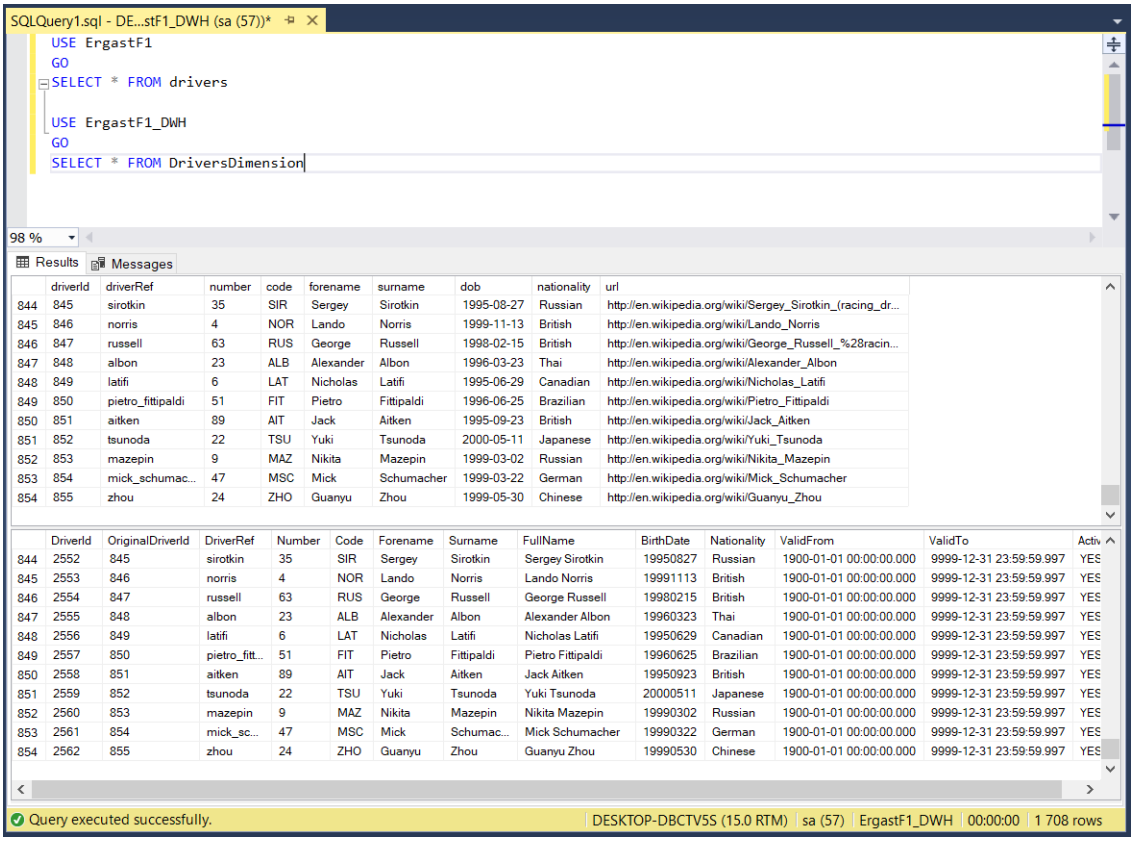
Naturalnym testem po załadowaniu danych do hurtowni będzie ich sprawdzenie i porównanie z tymi z tabel źródłowych. Testy miały na celu porównanie liczby rekordów w najważniejszych tabelach źródłowych zarówno przed 2022 rokiem jak i tych bieżących, z liczbą rekordów w tabelach w hurtowni.

W tym celu w narzędziu *Microsoft SQL Server Management Studio* uruchomione zostały zapytania języka sql widoczne na poniższych zrzutach ekranu.



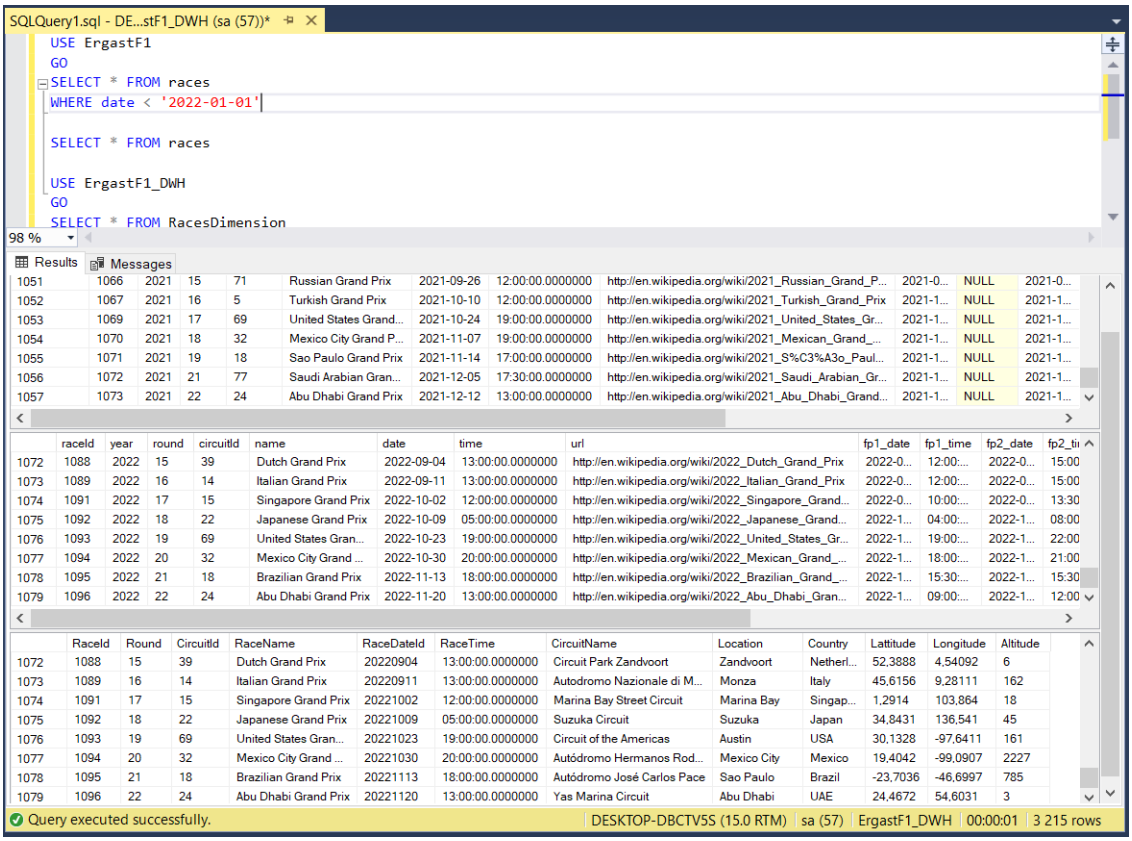
	constructorId	constructorRef	name	nationality	url
200	202	shadow-ford	Shadow-Ford	British	http://en.wikipedia.org/wiki/Shadow_Racing_Cars
201	203	shadow-matra	Shadow-M...	British	http://en.wikipedia.org/wiki/Shadow_Racing_Cars
202	204	brabham-alf...	Brabham-Al...	British	http://en.wikipedia.org/wiki/Brabham
203	205	lotus_racing	Lotus	Malaysian	http://en.wikipedia.org/wiki/Lotus_Racing
204	206	marussia	Marussia	Russian	http://en.wikipedia.org/wiki/Marussia_F1
205	207	caterham	Caterham	Malaysian	http://en.wikipedia.org/wiki/Caterham_F1
206	208	lotus_f1	Lotus F1	British	http://en.wikipedia.org/wiki/Lotus_F1
207	209	manor	Manor Mar...	British	http://en.wikipedia.org/wiki/Manor_Motorsport
208	210	haas	Haas F1 Te...	American	http://en.wikipedia.org/wiki/Haas_F1_Team
209	211	racing_point	Racing Point	British	http://en.wikipedia.org/wiki/Racing_Point_F1_Team
210	213	alphatauri	AlphaTauri	Italian	http://en.wikipedia.org/wiki/Scuderia_AlphaTauri
211	214	alpine	Alpine F1 T...	French	http://en.wikipedia.org/wiki/Alpine_F1_Team

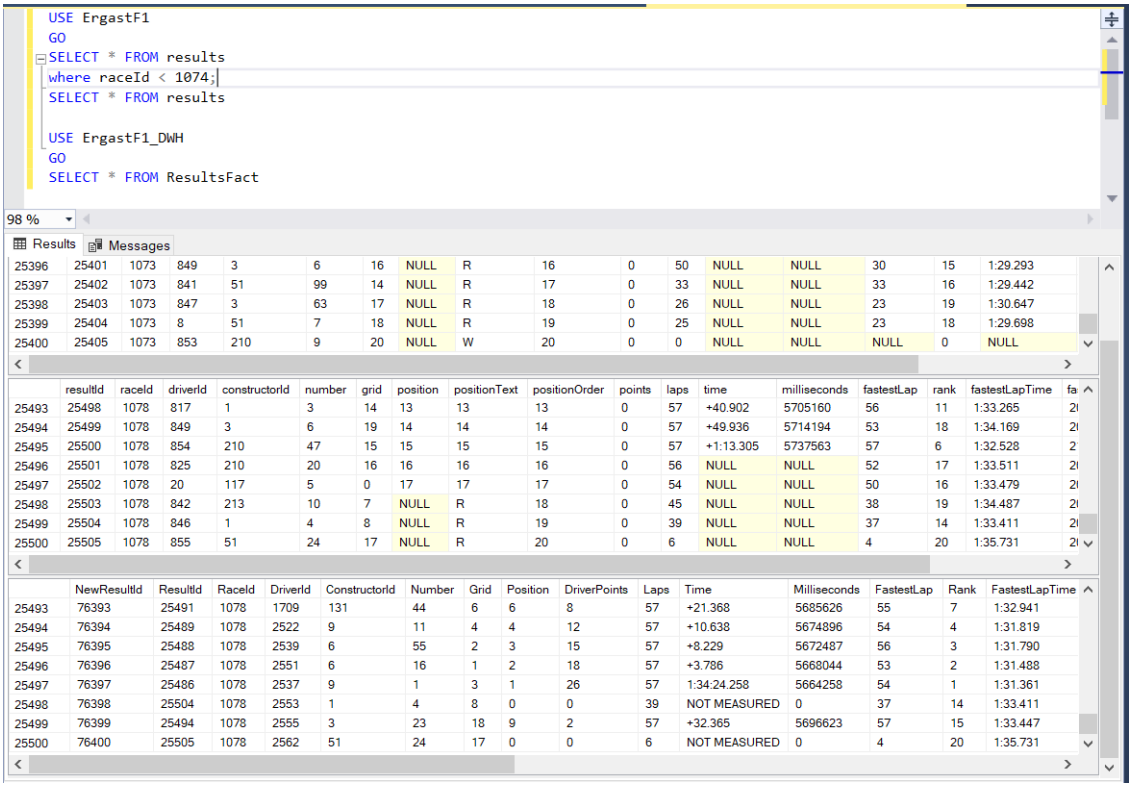
	ConstructorId	ConstructorRef	Name	Nationality
200	202	shadow-ford	Shadow-Ford	British
201	203	shadow-matra	Shadow-M...	British
202	204	brabham-alf...	Brabham-Al...	British
203	205	lotus_racing	Lotus	Malaysian
204	206	marussia	Marussia	Russian
205	207	caterham	Caterham	Malaysian
206	208	lotus_f1	Lotus F1	British
207	209	manor	Manor Mar...	British
208	210	haas	Haas F1 Te...	American
209	211	racing_point	Racing Point	British
210	213	alphatauri	AlphaTauri	Italian
211	214	alpine	Alpine F1 T...	French



	driverId	driverRef	number	code	forename	surname	dob	nationality	url
844	845	sirotkin	35	SIR	Sergey	Sirotkin	1995-08-27	Russian	http://en.wikipedia.org/wiki/Sergey_Sirotkin_(racing_dr...
845	846	norris	4	NOR	Lando	Norris	1999-11-13	British	http://en.wikipedia.org/wiki/Lando_Norris
846	847	russell	63	RUS	George	Russell	1998-02-15	British	http://en.wikipedia.org/wiki/George_Russell_%28racin...
847	848	albon	23	ALB	Alexander	Albon	1996-03-23	Thai	http://en.wikipedia.org/wiki/Alexander_Albon
848	849	latifi	6	LAT	Nicholas	Latifi	1995-06-29	Canadian	http://en.wikipedia.org/wiki/Nicholas_Latifi
849	850	pietro_fittipaldi	51	FIT	Pietro	Fittipaldi	1996-06-25	Brazilian	http://en.wikipedia.org/wiki/Pietro_Fittipaldi
850	851	aitken	89	AIT	Jack	Aitken	1995-09-23	British	http://en.wikipedia.org/wiki/Jack_Aitken
851	852	tsunoda	22	TSU	Yuki	Tsunoda	2000-05-11	Japanese	http://en.wikipedia.org/wiki/Yuki_Tsunoda
852	853	mazepin	9	MAZ	Nikita	Mazepin	1999-03-02	Russian	http://en.wikipedia.org/wiki/Nikita_Mazepin
853	854	mick_schumac...	47	MSC	Mick	Schumacher	1999-03-22	German	http://en.wikipedia.org/wiki/Mick_Schumacher
854	855	zhou	24	ZHO	Guanyu	Zhou	1999-05-30	Chinese	http://en.wikipedia.org/wiki/Guanyu_Zhou

	DriverId	OriginalDriverId	DriverRef	Number	Code	Forename	Surname	FullName	BirthDate	Nationality	ValidFrom	ValidTo	Activ
844	2552	845	sirotkin	35	SIR	Sergey	Sirotkin	Sergey Sirotkin	19950827	Russian	1900-01-01 00:00:00.000	9999-12-31 23:59:59.997	YES
845	2553	846	norris	4	NOR	Lando	Norris	Lando Norris	19991113	British	1900-01-01 00:00:00.000	9999-12-31 23:59:59.997	YES
846	2554	847	russell	63	RUS	George	Russell	George Russell	19980215	British	1900-01-01 00:00:00.000	9999-12-31 23:59:59.997	YES
847	2555	848	albon	23	ALB	Alexander	Albon	Alexander Albon	19960323	Thai	1900-01-01 00:00:00.000	9999-12-31 23:59:59.997	YES
848	2556	849	latifi	6	LAT	Nicholas	Latifi	Nicholas Latifi	19950629	Canadian	1900-01-01 00:00:00.000	9999-12-31 23:59:59.997	YES
849	2557	850	pietro_fitt...	51	FIT	Pietro	Fittipaldi	Pietro Fittipaldi	19960625	Brazilian	1900-01-01 00:00:00.000	9999-12-31 23:59:59.997	YES
850	2558	851	aitken	89	AIT	Jack	Aitken	Jack Aitken	19950923	British	1900-01-01 00:00:00.000	9999-12-31 23:59:59.997	YES
851	2559	852	tsunoda	22	TSU	Yuki	Tsunoda	Yuki Tsunoda	20000511	Japanese	1900-01-01 00:00:00.000	9999-12-31 23:59:59.997	YES
852	2560	853	mazepin	9	MAZ	Nikita	Mazepin	Nikita Mazepin	19990302	Russian	1900-01-01 00:00:00.000	9999-12-31 23:59:59.997	YES
853	2561	854	mick_sc...	47	MSC	Mick	Schumac...	Mick Schumacher	19990322	German	1900-01-01 00:00:00.000	9999-12-31 23:59:59.997	YES
854	2562	855	zhou	24	ZHO	Guanyu	Zhou	Guanyu Zhou	19990530	Chinese	1900-01-01 00:00:00.000	9999-12-31 23:59:59.997	YES





Analizując zrzuty ekranu możemy dojść do wniosku, że liczność wierszy w tabelach źródłowych oraz w hurtowni jest identyczna. Również wartości tych samych zmiennych dla ostatnich rekordów pokrywają się w poszczególnych tabelach.

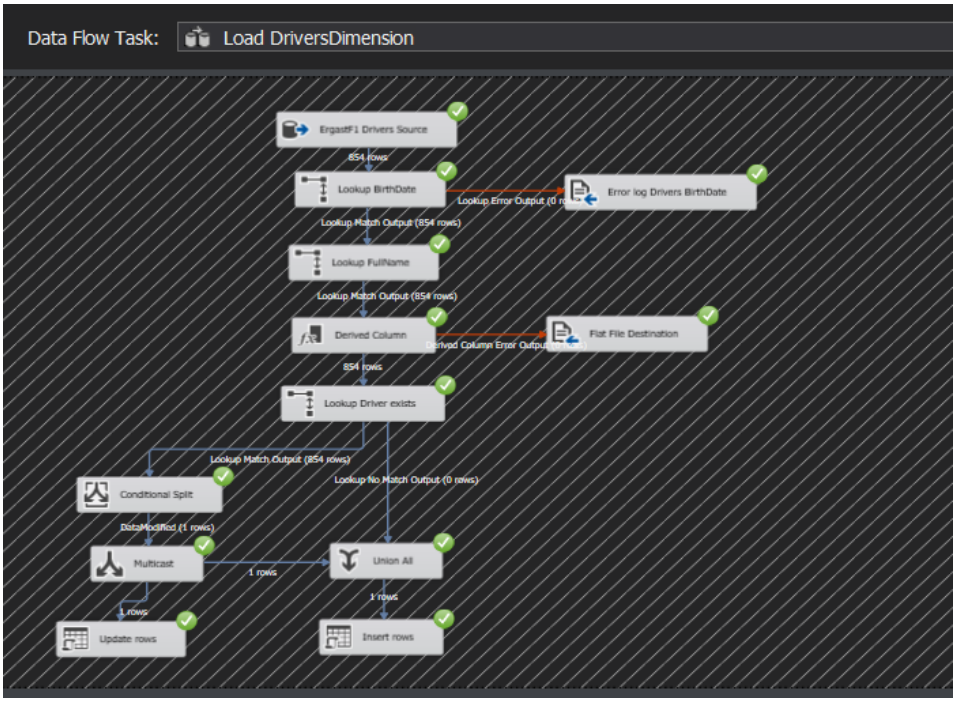
W celu sprawdzenia poprawności załadowania tabel, które były zasilone nowymi rekordami z danymi z 2022 roku, możemy również zauważyć, że liczba rekordów sprzed 2022 roku i rekordów bieżących sumuję się do wymiaru hurtowni, jak również zgadza się z wynikami procesu ETL (patrząc na tabelę z wyścigami, łatwo zauważyć, że ostatnie ID wyścigu z poprzedniego roku jest równe 1073 - z tego korzystamy przy zapytaniach języka sql)

9.3 Sprawdzenie działania implementacji Slowly Changing Dimension

Sprawdzimy w tej sekcji poprawność działania zaimplementowanej techniki *Slowly changing dimension*. W tym celu zaktualizujemy rekord w bazie źródłowej *drivers* następująco:

Update drivers SET nationality = 'British' where driverId=840

Sprawdźmy procesowanie ETL tabeli źródłowej po tej zmianie (oczywiście dane z tej tabeli przez aktualizacją zostały załadowane już do hurtowni). Poniżej przedstawiony jest schemat ETL:



Widzimy, że jeden z wierszy został zaktualizowany oraz jeden nowy wiersz został dodany. Sprawdźmy jak wygląda tabela wymiarowa za pomocą zapytania typu *SELECT*.

Widzimy, że zmiana została wprowadzona poprawnie z wymaganiami dotyczącymi omawianej techniki. Został dodany nowy rekord, z nową wartością *DriverId* i zachowaną *OriginalDriverId*.

Dodatkowo dla rekordu przed wprowadzoną zmianą, wprowadzono aktualizację zmiennej *ValidTo* na datę zasilenia hurtowni zauktualizowanym wierszem. Taką samą wartość przyjmuje kolumna *ValidFrom* nowego rekordu.

```
select * from DriversDimension
```

98 %

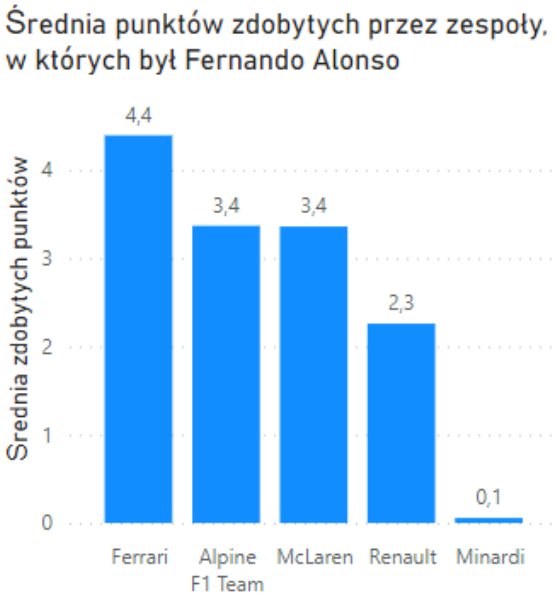
Results | Messages

	DriverId	OriginalDriverId	DriverRef	Number	Code	Forename	Surname	FullName	BirthDate	Nationality	ValidFrom	ValidTo
832	2540	833	merhi	98	MER	Roberto	Merhi	Roberto Merhi	19910322	Spanish	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
833	2541	834	rossi	53	RSS	Alexander	Rossi	Alexander Rossi	19910925	American	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
834	2542	835	jolyon_palmer	30	PAL	Jolyon	Palmer	Jolyon Palmer	19910120	British	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
835	2543	836	wehrlein	94	WEH	Pascal	Wehrlein	Pascal Wehrlein	19941018	German	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
836	2544	837	haryanto	88	HAR	Rio	Haryanto	Rio Haryanto	19930122	Indonesi...	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
837	2545	838	vandoorne	2	VAN	Stoffel	Vandoorne	Stoffel Vandoorne	19920326	Belgian	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
838	2546	839	ocon	31	OCO	Esteban	Ocon	Esteban Ocon	19960917	French	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
839	2547	840	stroll	18	STR	Lance	Stroll	Lance Stroll	19981029	Canadian	1900-01-01 00:00:00.000	2022-06-20 15:30:25.00
840	2548	841	giovinazzi	99	GIO	Antonio	Giovinazzi	Antonio Giovinazzi	19931214	Italian	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
841	2549	842	gasly	10	GAS	Pierre	Gasly	Pierre Gasly	19960207	French	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
842	2550	843	brendon_hartley	28	HAR	Brendon	Hartley	Brendon Hartley	19891110	New Zea...	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
843	2551	844	leclerc	16	LEC	Charles	Leclerc	Charles Leclerc	19971016	Monega...	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
844	2552	845	sirotkin	35	SIR	Sergey	Sirotkin	Sergey Sirotkin	19950827	Russian	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
845	2553	846	norris	4	NOR	Lando	Norris	Lando Norris	19991113	British	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
846	2554	847	russell	63	RUS	George	Russell	George Russell	19980215	British	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
847	2555	848	albon	23	ALB	Alexander	Albon	Alexander Albon	19960323	Thai	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
848	2556	849	latifi	6	LAT	Nicholas	Latifi	Nicholas Latifi	19950629	Canadian	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
849	2557	850	pietro_fittipaldi	51	FIT	Pietro	Fittipaldi	Pietro Fittipaldi	19960625	Brazilian	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
850	2558	851	aitken	89	AIT	Jack	Aitken	Jack Aitken	19950923	British	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
851	2559	852	tsunoda	22	TSU	Yuki	Tsunoda	Yuki Tsunoda	20000511	Japanese	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
852	2560	853	mazepin	9	MAZ	Nikita	Mazepin	Nikita Mazepin	19990302	Russian	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
853	2561	854	mick_schumac...	47	MSC	Mick	Schumacher	Mick Schumacher	19990322	German	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
854	2562	855	zhou	24	ZHO	Guanyu	Zhou	Guanyu Zhou	19990530	Chinese	1900-01-01 00:00:00.000	9999-12-31 23:59:59.99
855	2563	840	stroll	18	STR	Lance	Stroll	Lance Stroll	19981029	British	2022-06-20 15:30:26.000	9999-12-31 23:59:59.99

Query executed successfully. | DESKTOP-DBCTV5S (15.0 RTM) | sa (57) | ErgastF1_DWH | 00:00:00 | 855 rows

9.4 Porównanie wyników na raportach i z kwerend

Chcemy się upewnić, że narzędzie raportujące wykonuje to co chcemy. W tym celu porównano wyniki kwerend SQL z kilkoma wynikami na raportach.



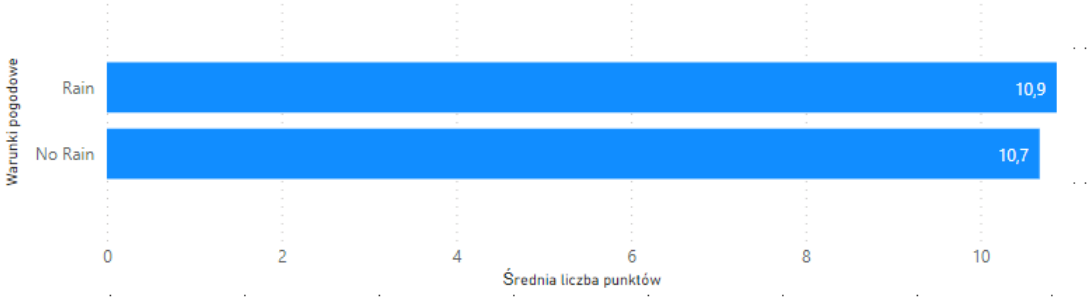
```sql
SELECT c.Name, AVG(DriverPoints) as AvgPoints
FROM ResultsFact f
JOIN ConstructorsDimension c
ON c.ConstructorId = f.ConstructorId
WHERE c.ConstructorRef
    in ('ferrari', 'alpine', 'mclaren', 'renault', 'minardi')
GROUP BY c.Name
```

80 %

Results | Messages

	Name	AvgPoints
1	Alpine F1 Team	3,36666666666667
2	Ferrari	4,39062529176515
3	McLaren	3,3585690515807
4	Minardi	0,056547619047619
5	Renault	2,2579415501906

Średnia liczba punktów uzyskanych przez Sebastiana Vettela z podziałem na warunki pogodowe




```sql
SELECT d.FullName, AVG(DriverPoints) as AvgPoints
FROM ResultsFact f
JOIN DriversDimension d
ON d.DriverId = f.DriverId
WHERE d.FullName = 'Sebastian Vettel' AND f.Precipitation > 0
GROUP BY d.FullName
UNION
SELECT d.FullName, AVG(DriverPoints) as AvgPoints
FROM ResultsFact f
JOIN DriversDimension d
ON d.DriverId = f.DriverId
WHERE d.FullName = 'Sebastian Vettel' AND f.Precipitation = 0
GROUP BY d.FullName
```

0 %

Results | Messages

	FullName	AvgPoints
1	Sebastian Vettel	10,7513812154696
2	Sebastian Vettel	10,873786407767

Grid grouped	Średnia elementów Grid	PositionFinished
11-15	12,99	9,22
1-5	2,31	3,81
16-	19,86	11,79
6-10	8,00	6,76
Suma	**11,16**	**7,96**

```sql
SELECT GridGrouped, AVG(CAST(Position AS FLOAT))
FROM (
    SELECT Position, Grid,
    CASE
        WHEN Grid >= 1 AND Grid <= 5 THEN '1-5'
        WHEN Grid >= 6 AND Grid <= 10 THEN '6-10'
        WHEN Grid >= 11 AND Grid <= 15 THEN '11-15'
        ELSE '16-'
    END AS GridGrouped
    FROM ResultsFact f
    WHERE Position > 0
) AS t1
GROUP BY GridGrouped
```

Results Messages

	GridGrouped	(No column name)
1	6-10	6,76269149894863
2	1-5	3,73085221143474
3	16-	11,7988324008757
4	11-15	9,22078342505665

Jak widzimy na przykładach, otrzymujemy te same wyniki w obydwu przypadkach co świadczy o tym, że wyniki otrzymywane w narzędziu raportującym są właściwe.

10 Podsumowanie

Podsumowując, w ramach projektu powstało rozwiązanie Business Intelligence umożliwiające potencjalnemu klientowi sprawne raportowanie i analizę historycznych danych wyścigów Formuły 1.

Jest to możliwe przede wszystkim dzięki następującym rozwiązaniom:

- Model hurtowni danych - model gwiazdy z tabelą faktów zawierającą wyniki z poszczególnych wyścigów oraz z tabelami wymiarowymi umożliwiającymi ich późniejszą analizę pod kątem zawodników, wyścigów/torów wyścigowych czy konstruktorów.
- Zaimplementowany w SSIS proces ETL umożliwiający sprawne zasilanie hurtownię danymi dobrej jakości (tzn. dane w tym procesie są przetworzone tak, aby były czytelne dla użytkownika biznesowego). Poza tym, proces ten umożliwia sprawne wprowadzanie zamian w danych o danym zawodniku, zachowując przy tym jego poprzednie wartości (dzięki technice slowly changing dimension).
- Model danych z hurtowni importowany do systemu BI (Power BI) w formie umożliwiającej sprawne raportowanie. Zdefiniowana hierarchia oraz obróbka danych umożliwiają raportowanie na różnych poziomach szczegółowości. Przygotowane są również przykładowe raporty:
  - Wpływ zespołu na wyniki osiągane przez Fernando Alonso.
  - Popularność Formuły 1 (pod względem liczby wyścigów i kierowców) w poszczególnych krajach.

– Wpływ opadów atmosferycznych w dniu wyścigu na osiągane wyniki Sebasiana Vettela.
– Porównanie pozycji startowej oraz pozycji po kwalifikacjach z końcową pozycją zajmowaną przez zawodników w wyścigach.

Z punktu widzenia biznesu, może to ułatwić odpowiednim zespołom przygotować taktykę dla zawodników na przyszły sezon, np. kładąc większy nacisk na starty w kwalifikacjach czy dostosowując swoje szanse na starty w wyścigach przy opadach lub bez opadów.

Dodatkowo zespoły mogą rozwinąć swoją strategię marketingową w krajach, w których odbywa się najwięcej wyścigów czy odnotowano najwięcej rezultatów. W krajach o największej popularności Formuły 1 zespoły mają większe szanse znaleźć młodych kierowców z największym potencjałem.