Formula 1 performance in different weather conditions

Barbara Seweryn, Michał Wietecki

June 11, 2025

1 Introduction

1.1 Business case

The purpose of this project is developing a data warehouse containing Formula 1 data for performance analysis depending on weather conditions.

1.2 Detailed objectives

1. Comparison of team and driver efficiency on various tracks in various weather conditions.
2. Pattern analysis regarding team strategy and weather.
3. Providing easy access to historical data exploration in one integrated source.

1.3 Benefits of our solution

1. **For constructors** - optimization of race strategies, better race preparation depending on weather conditions, performance analysis of their drivers
2. **For sport reporters** - ability to compare teams and drivers based on their strategies and generate visualizations, easy creation of reports and summaries.
3. **For F1 fans** - interactive exploration of historical race data, possibility to predict future race results – for example for betting purposes

2 Data sources - first attempt

1. **Open Meteo Historical Weather Data**
   Historical hourly weather data, taking Longitude, Latitude and time as parameters and returns weather information like rain, wind speed, humidity, temperature etc.
   https://open-meteo.com/en/docs/historical-weather-api
2. **Open Formula 1 Ergast API**
   Returns data about drivers, race results, circuits and more.
   https://ergast.com/mrd/
3. **Wikipedia circuit information table**
   Contains circuit name, location, years it was active etc.
   https://en.wikipedia.org/wiki/List_of_Formula_One_circuits

3 Data sources - final attempt

Unfortunately during the development of the project the Ergast API shut down and we could no longer use it.
We had to find another data source and we decided to use this data from GitHub:
https://github.com/toUpperCase78/formula1-datasets/tree/master

Here we found individual csv files containing race results, driver data and season calendars from 2020 to 2024.
We decided to use this data and narrow down our project to these seasons.
Here a sample of the race results data is shown:

1	Track	Position	No	Driver	Team	Starting Grid	Laps	Time/Retired	Points	+1 Pt	Fastest Lap
2	Bahrain	1	16	Charles Leclerc	Ferrari	1	57	1:37:33.584	26	Yes	1:34.570
3	Bahrain	2	55	Carlos Sainz	Ferrari	3	57	+5.598	18	No	1:35.740
4	Bahrain	3	44	Lewis Hamilton	Mercedes	5	57	+9.675	15	No	1:36.228
5	Bahrain	4	63	George Russell	Mercedes	9	57	+11.211	12	No	1:36.302
6	Bahrain	5	20	Kevin Magnussen	Haas Ferrari	7	57	+14.754	10	No	1:36.623
7	Bahrain	6	77	Valtteri Bottas	Alfa Romeo Ferrari	6	57	+16.119	8	No	1:36.599
8	Bahrain	7	31	Esteban Ocon	Alpine Renault	11	57	+19.423	6	No	1:37.110
9	Bahrain	8	22	Yuki Tsunoda	AlphaTauri RBPT	16	57	+20.386	4	No	1:37.104
10	Bahrain	9	14	Fernando Alonso	Alpine Renault	8	57	+22.390	2	No	1:36.733
11	Bahrain	10	24	Guanyu Zhou	Alfa Romeo Ferrari	15	57	+23.064	1	No	1:36.685

Figure 1: Race result data example

Here is a sample of the season calendar (which will be used in populating the FactRaceWeather table):

Round	Race Date	GP Name	Country	City	Circuit Name	First GP	Number of Laps	Circuit Length(km)	Race Distance(km)	Lap Record	Record Owner	Rec
1	20/03/2022	Gulf Air Bahrain GP	Bahrain	Sakhir	Bahrain International Circuit	2004	57	5.412	308.238	1:31.447	Pedro de la Rosa	200
2	27/03/2022	STC Saudi Arabian GP	Saudi Arabia	Jeddah	Jeddah Corniche Circuit	2021	50	6.174	308.45	1:30.774	Lewis Hamilton	202
3	10/04/2022	Heineken Australian GP	Australia	Melbourne	Albert Park Circuit	1996	58	5.278	306.124	1:20.260	Charles Leclerc	202
4	24/04/2022	Rolex Gran Premio Del Made in Italy e Dell'Emilia-Romagna	Italy	Imola	Autodromo Enzo e Dino Ferrari	1980	63	4.909	309.049	1:15.484	Lewis Hamilton	202
5	08/05/2022	Crypto.com Miami GP	United States	Miami	Miami International Autodrome	2022	57	5.412	308.326	1:31.361	Max Verstappen	202
6	22/05/2022	Pirelli Gran Premio de España	Spain	Catalunya	Circuit de Barcelona-Catalunya	1991	66	4.675	308.424	1:18.149	Max Verstappen	202
7	29/05/2022	Grand Prix de Monaco	Monaco	Monte Carlo	Circuit de Monaco	1950	78	3.337	260.286	1:12.909	Lewis Hamilton	202
8	12/06/2022	Azerbaijan GP	Azerbaijan	Baku	Baku City Circuit	2016	51	6.003	306.049	1:43.009	Charles Leclerc	201
9	19/06/2022	AWS Grand Prix du Canada	Canada	Montréal	Circuit Gilles-Villeneuve	1978	70	4.361	305.27	1:13.078	Valtteri Bottas	201

Figure 2: Season calendar sample data

4 Data Warehouse structure

4.1 Table diagram

In this phase of the project, we created a database in SQL Server, created all the tables and the diagram. In the later part of the project, this database was the destination of our data loading.

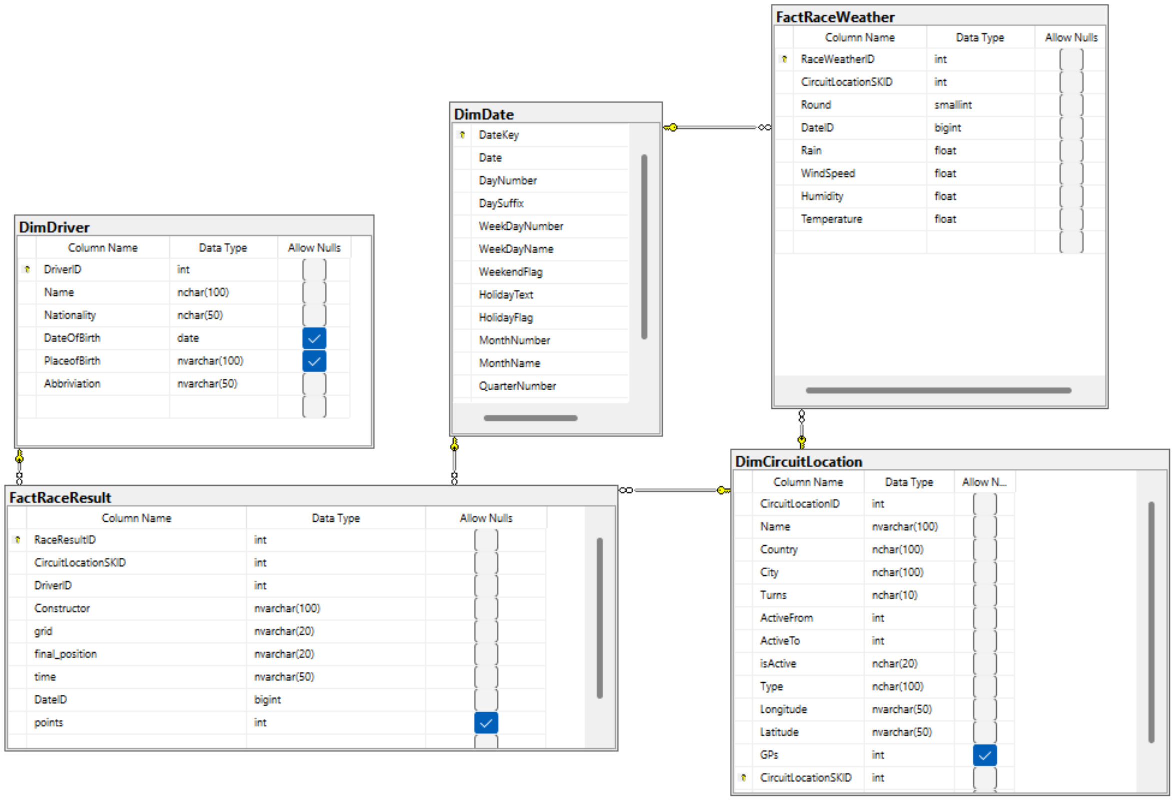


Figure 3: Data warehouse diagram

4.2 Table descriptions

In this section all of the tables and their columns are described.

4.2.1 FactRaceResult

- RaceResultID - primary key
- CircuitLocationID - foreign key referencing the DimCircuitLocation table
- DriverID - foreign key referencing the DimDriver table
- Constructor - name of the Constructor (team)
- grid - driver's staring position for the race
- final position - driver's final position in the race
- status - Finished/Collision/+1 Lap etc.
- time - race time for the winner, +... for the 9 following drivers, empty for others
- DateID - foreign key, referencing the DimDate table

4.2.2 FactRaceWeather

- RaceWeatherID - primary key
- CircuitLocationID - foreign key referencing the DimCircuitLocation table
- Season - year of the race
- Round - race number in the season (1 for the first race, 2 for the second etc.)

- DateID - foreign key referencing the DimDate table
- SumRain - sum of rain precipitation during the race, in mm
- AvgWindSpeed - average wind speed, in km/h
- AvgHumidity - average humidity
- AvgTemperature - in Celcius

4.2.3 DimDriver

- DriverID - primary key
- Name - name and surname of the driver
- Nationality
- DOB - date of birth

4.2.4 DimCircuitLocation

- SKID
- CircuitLocationID - primary key
- Name - circuit name
- Country
- City
- Turns - number of turns in a circuit
- ActiveFrom, ActiveTo, isActive - SCD2, dated referencing the DimDate table
- Type - circuit type (street, track)
- Longitude, Latitude - coordinates for the city

4.2.5 DimDate

Classic DimDate table for years 2020-2024.

5 ETL process - how will we populate the tables? - attempt 1

5.1 FactRaceResult

Every column (other than the primary and foreign key) will be extracted from Ergast Result API. (https://ergast.com/mrd/methods/results/)

Query Details

Series	Season	Round	Results
f1	2008	5	20

2008 Turkish Grand Prix

Race Results

Pos	No	Driver	Constructor	Laps	Grid	Time	Status	Points
1	2	Felipe Massa	Ferrari	58	1	1:26:49.451	Finished	10
2	22	Lewis Hamilton	McLaren	58	3	+3.779	Finished	8
3	1	Kimi Räikkönen	Ferrari	58	4	+4.271	Finished	6
4	4	Robert Kubica	BMW Sauber	58	5	+21.945	Finished	5
5	3	Nick Heidfeld	BMW Sauber	58	9	+38.741	Finished	4
6	5	Fernando Alonso	Renault	58	7	+53.724	Finished	3
7	10	Mark Webber	Red Bull	58	6	+1:04.229	Finished	2

Figure 4: Race result Ergast API data

This table should be updated every time there is a race. Races are from 1 to 5 weeks apart, so the table should be updated once a week.

5.2 FactRaceWeather

For this table, we will use the circuit coordinates from the DimCircuitLocation table and the StartHour FactRaceWeather.
Using these parameters we can get hourly weather data (a race usually lasts about 2 hours) and aggregate the 2 hourly weather records, so we get a RaceWeather row.
To do this we will calculate the average of those two values for WindSpeed, Humidity, and Temperature, and sum for Rain.
The start hour will be extracted from Race table in Egast API.
https://ergast.com/api/f1/2005

```
Coordinates 52.5483283996582°N 13.407821655273438°E
Elevation 38.0 m asl
Timezone NoneNone
Timezone difference to GMT+0 0 s
                         date  temperature_2m  rain  wind_speed_100m  \
0   2025-04-24 00:00:00+00:00        9.958500   0.0        24.456827
1   2025-04-24 01:00:00+00:00       10.158501   0.0        25.056231
2   2025-04-24 02:00:00+00:00        9.508500   0.0        25.928123
3   2025-04-24 03:00:00+00:00        9.058500   0.0        25.455843
4   2025-04-24 04:00:00+00:00        8.558500   0.0        24.316660
..                        ...             ...   ...              ...
355 2025-05-08 19:00:00+00:00       13.708500   0.0         8.707237
356 2025-05-08 20:00:00+00:00       12.458500   0.0        11.236671
357 2025-05-08 21:00:00+00:00       11.358500   0.0        14.843180
358 2025-05-08 22:00:00+00:00       10.458500   0.0        14.589996
359 2025-05-08 23:00:00+00:00        9.258500   0.0        10.373061

     relative_humidity_2m
0               85.321640
1               81.360878
2               81.834877
3               83.197502
4               84.584442
..                    ...
355             36.436878
356             43.297363
357             47.917053
358             49.064167
359             54.729885

[360 rows x 5 columns]
```

Figure 5: Weather Data example imported in .ipynb

This table should be updated every time there is a race. Races are from 1 to 5 weeks apart, so the table should be updated once a week.

5.3 DimCircuitLocation

This table will be populated by the Wikipedia database. (other than the primary and foreign key)

The only problem we will have here is that from that table we get the City and Country where the circuit is. However, to fetch data from the Weather API, we need specific Longitude and Latitude of this city.

To do this we can either find data manually and create another table with cities and their coordinates or use AI assistance to handle it for us. Right now we are not sure how to handle that problem, so for now we will assume that we have the coordinates as well as the City and Country name.

We will also have to transform the column that says what years was the circuit active to a format fitting to SCD2.

Formula One circuits

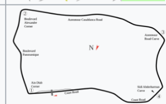
Circuit	Map	Type	Direction	Location	Country	Last length used	Turns	Grands Prix	Season(s)	Grands Prix held
Adelaide Street Circuit		Street circuit	Clockwise	Adelaide	Australia	3.780 km (2.349 mi)	16	Australian Grand Prix	1985–1995	11
Ain-Diab Circuit		Road circuit	Clockwise	Casablanca	Morocco	7.618 km (4.734 mi)	18	Moroccan Grand Prix	1958	1
Aintree Motor Racing Circuit		Road circuit	Clockwise	Aintree	United Kingdom	4.828 km (3.000 mi)	12	British Grand Prix	1955, 1957, 1959, 1961–1962	5

Figure 6: Circuit data example from Wikipedia

This table rarely changes, and if there are some changes (adding or removing a circuit from the season) they are done once a year, before the season starts. So this table could be updated only once a year.

5.4 DimDriver

This table will be populated fully using the Ergast Driver API. (other than the primary and foreign key)
(https://ergast.com/mrd/methods/drivers/)

Query Details

Series	Page	Results
f1	1 of 29	861

Driver Table

Driver Name	Permanent Number	Nationality	DOB	Information
Carlo Abate		Italian	1932-07-10	Biography
George Abecassis		British	1913-03-21	Biography
Kenny Acheson		British	1957-11-27	Biography
Philippe Adams		Belgian	1969-11-19	Biography
Walt Ader		American	1913-12-15	Biography
Kurt Adolff		German	1921-11-05	Biography

Figure 7: Driver data example from Ergast API

This table rarely changes, and if there are some changes they are mostly before the season starts or sometimes during the season. For safety, we could update this table once a month.

6 ETL process - how did we actually populate the tables? - final attempt

Again, because the Ergast API has shut down we had to completely change our approach when it comes to the ETL process.

Since the data that we worked with was complicated and not of best quality and the transformation that we planned to do also were quite demanding, we decided that the extraction and transformation of the data will be performed using python. We needed more control over the transformations and thats why we thought that choosing python is the better solution. Additionaly this way we could document

the transformation by putting it on our GitHub repository. After completing these processes we generated csv files and loaded then into the data warehouse using SSIS, loading the data from a flat file source (other then the DimDate table which we populated using a sql script).
Here is how we populated each table.

6.1 DimCircuitLocation

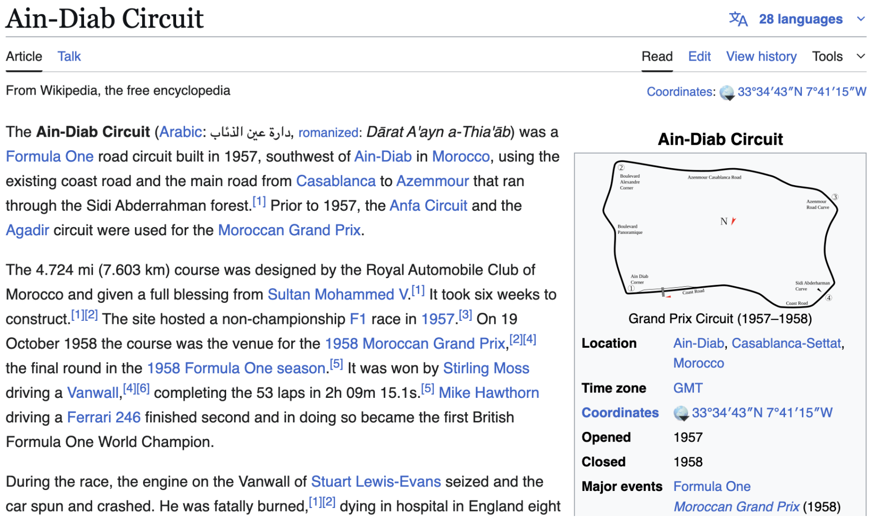
Starting off with a dimension table containing information about circuits, we used data from a Wikipedia page, as it was neatly stored and had most of the needed columns:

Formula One circuits

Circuit	Map	Type	Direction	Location	Country	Last length used	Turns	Grands Prix	Season(s)	Grands Prix held
Adelaide Street Circuit		Street circuit	Clockwise	Adelaide	Australia	3.780 km (2.349 mi)	16	Australian Grand Prix	1985–1995	11
Ain-Diab Circuit		Road circuit	Clockwise	Casablanca	Morocco	7.618 km (4.734 mi)	18	Moroccan Grand Prix	1958	1
Aintree Motor Racing Circuit		Road circuit	Clockwise	Aintree	United Kingdom	4.828 km (3.000 mi)	12	British Grand Prix	1955, 1957, 1959, 1961–1962	5

Figure 8: Wikipedia's "List of Formula One circuits"

We wrote a script using *BeautifulSoup* library in Python, which enabled us to extract the whole table to a .csv file. Additionally, we wanted to scrape exact geographical coordinates, for the purpose of getting weather data from locations of the circuits. However, the table shown above didn't store the coordinates. To extract them, we needed to get into a distinct site for each race, which could be found in the *Circuit* column. To achieve that, we used the *Selenium* library to create headless browsers. This was necessary, as the data on each race's site was dynamically rendered through *javascript*. Exemplary site (a) and the code used to extract coordinates from it (b) are shown below:



Ain-Diab Circuit

28 languages

Article Talk | Read Edit View history Tools

From Wikipedia, the free encyclopedia

Coordinates: 33°34′43″N 7°41′15″W

The **Ain-Diab Circuit** (Arabic: دارة عين الذئاب, romanized: *Dārat A'ayn a-Thia'āb*) was a Formula One road circuit built in 1957, southwest of Ain-Diab in Morocco, using the existing coast road and the main road from Casablanca to Azemmour that ran through the Sidi Abderrahman forest.[1] Prior to 1957, the Anfa Circuit and the Agadir circuit were used for the Moroccan Grand Prix.

The 4.724 mi (7.603 km) course was designed by the Royal Automobile Club of Morocco and given a full blessing from Sultan Mohammed V.[1] It took six weeks to construct.[1][2] The site hosted a non-championship F1 race in 1957.[3] On 19 October 1958 the course was the venue for the 1958 Moroccan Grand Prix,[2][4] the final round in the 1958 Formula One season.[5] It was won by Stirling Moss driving a Vanwall,[4][6] completing the 53 laps in 2h 09m 15.1s.[5] Mike Hawthorn driving a Ferrari 246 finished second and in doing so became the first British Formula One World Champion.

During the race, the engine on the Vanwall of Stuart Lewis-Evans seized and the car spun and crashed. He was fatally burned,[1][2] dying in hospital in England eight

Ain-Diab Circuit

Grand Prix Circuit (1957–1958)

Location	Ain-Diab, Casablanca-Settat, Morocco
Time zone	GMT
Coordinates	33°34′43″N 7°41′15″W
Opened	1957
Closed	1958
Major events	Formula One *Moroccan Grand Prix* (1958)

(a) Wikipedia: "Ain-Diab Circuit"

```python
def extract_coordinates(current_row):
    link_tag = current_row[0].find("a")
    if link_tag and "href" in link_tag.attrs:
        link = link_tag["href"]
        full_link = "https://en.wikipedia.org" + link
    else:
        print("error")

    driver = webdriver.Chrome()
    driver.get(full_link)

    soup = BeautifulSoup(driver.page_source, "html.parser")
    table = soup.find("table", class_="infobox")

    # some coordinates are in the infobox and some outside it in a span with class geo-dec
    coordinates_row = table.find("th", string=lambda text: text and "Coordinates" in text)
    if coordinates_row:
        td = coordinates_row.find_next_sibling("td")
        latitude = td.find("span", class_="latitude")
        latitude = latitude.text
        longitude = td.find("span", class_="longitude")
        longitude = longitude.text
    else:
        geo_dec = soup.find("span", class_="geo-dec")
        latitude, longitude = decimal_to_dms(geo_dec.text)

    driver.quit()

    if latitude and longitude:
        return [latitude, longitude]
    else:
        return [0,0]
```

(b) Function used to extract coordinates from a single race site

Figure 9: Overview of circuit location and coordinate extraction method

This way, we extracted all the needed data, and the transformation part began. Some of the things we did were:

1. The table was transformed using the Slowly Changing Dimension Type 2 (SCD2) technique by generating multiple rows for each circuit, based on the active season years indicated in the 'Season(s)' column. The difference from the standatd approach is the fact that 'isActive' column indicates, wheather the circuit's cadency is an active one.
2. Converted the extracted coordinates to a format accepted by the weather api (decimal).
3. Generated circuit IDs as well as surrogate keys.

The table from the .csv file passed to the loading stage looked like this:

	Circuit	Type	Direction	Location	Country	Last length used	Turns	Latitude	Longitude	Grands Prix held	from	to	isActivePeriod	CircuitLocationID	SKID
0	Adelaide Street Circuit	Street circuit	Clockwise	Adelaide	Australia	3.780 km (2.349 mi)	16	-34.9306	138.6206	11	1985	1995	False	46946	1
1	Ain-Diab Circuit	Road circuit	Clockwise	Casablanca	Morocco	7.618 km (4.734 mi)	18	33.5786	-7.6875	1	1958	1958	False	20953	2
2	Aintree Motor Racing Circuit	Road circuit	Clockwise	Aintree	United Kingdom	4.828 km (3.000 mi)	12	53.4769	-2.9406	5	1955	1955	False	26284	3

Figure 10: Table passed to the loading stage as DimCircuitLocation

6.2 FactRaceWeather

The base for this table was a concatenation of 5 different data frames from our GitHub source - one for each year. They stored basic data about every race that took place in the span of 5 years. The feature that was the most important for us was the date of the race. A part of one of those tables is shown below:

	Round	Country	City	Circuit Name	GP Name	Race Date	First GP	Number of Laps	Circuit Length(km)	Race Distance(km)	Lap Record	Record Owner	Record Year	Turns
0	1	Bahrain	Sakhir	Bahrain International Circuit	Gulf Air Bahrain GP	20/03/2022	2004	57	5.412	308.238	1:31.447	Pedro de la Rosa	2005	15
1	2	Saudi Arabia	Jeddah	Jeddah Corniche Circuit	STC Saudi Arabian GP	27/03/2022	2021	50	6.174	308.450	1:30.774	Lewis Hamilton	2021	27
2	3	Australia	Melbourne	Albert Park Circuit	Heineken Australian GP	10/04/2022	1996	58	5.278	306.124	1:20.260	Charles Leclerc	2022	14

Figure 11: First rows of the table containing data from 2022

After normalizing the structure of all five tables - enabling them to be concatenated — we tried to join the resulting table with the previously created DimCircuitLocation to retrieve the latitude and longitude needed for subsequent weather data extraction. Unfortunately, the join was impossible, which was a result of a number of spelling mistakes in the new data. We were forced to transfer the circuit ID numbers manually.

After cleaning the data, we were able to gather all columns needed to perform the extraction of meteorological data for each race's exact location and date.

Iterating through the dataset, we were able to extract the daily weather data (mean temperature, maximum wind speed, mean humidity, and rain sum) from the Open Meteo Weather API. To achieve that, we used a Python script, generated by the Open Meteo Weather API website. Here we decided to use daily weather information instead of hourly because we had a lot of difficulties when looking for race start hour data.

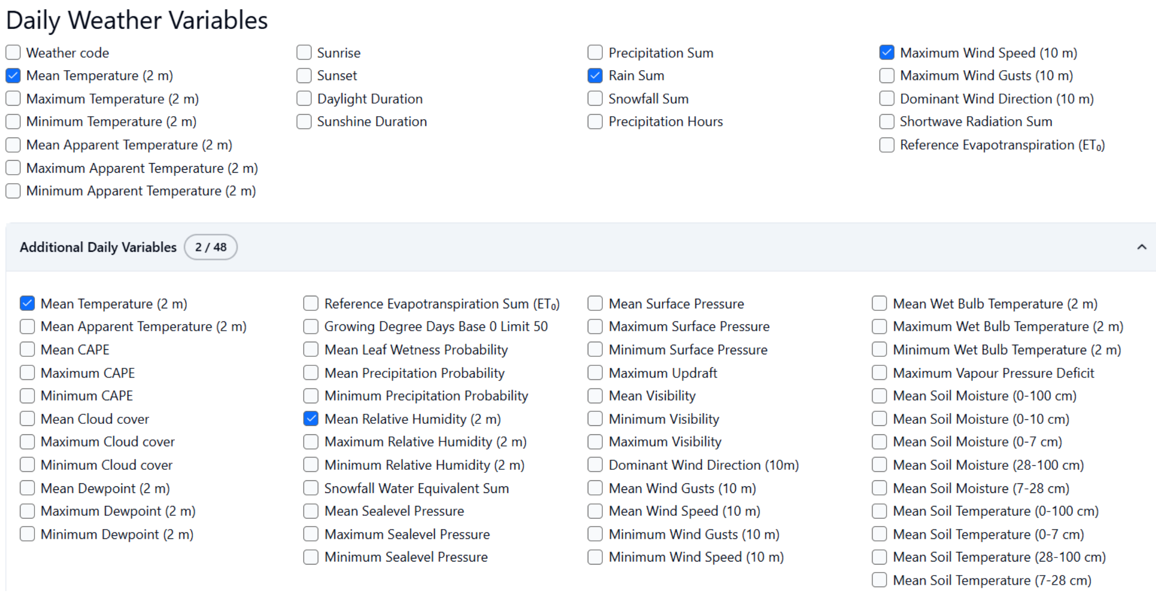


Figure 12: Open Meteo website

Usage

```python
import openmeteo_requests

import pandas as pd
import requests_cache
from retry_requests import retry

# Setup the Open-Meteo API client with cache and retry on error
cache_session = requests_cache.CachedSession('.cache', expire_after = -1)
retry_session = retry(cache_session, retries = 5, backoff_factor = 0.2)
openmeteo = openmeteo_requests.Client(session = retry_session)

# Make sure all required weather variables are listed here
# The order of variables in hourly or daily is important to assign them correctly below
url = "https://archive-api.open-meteo.com/v1/archive"
params = {
    "latitude": 52.52,
    "longitude": 13.41,
    "start_date": "2025-05-25",
    "end_date": "2025-06-08",
    "daily": ["temperature_2m_mean", "rain_sum", "wind_speed_10m_max", "relative_humidity_2m_mean"],
    "timezone": "GMT"
}
```

Figure 13: Code generated by the Open Meteo website

After cleaning the data (letting go of some columns), transforming it (e.g. converting the date column to a Style112 format used in the DimDate table) and handling a newly discovered problem (two races had the same date - one of those and one other had to be updated with a correct date), the table is ready to be passed into the SSIS:

	Race Date	CircuitLocationID_x	Round	Latitude	Longitude	temperature	wind	rain	humidity	RaceWeatherSKID
0	20200705	174	1	47.2197	14.7647	19.6	8.9	0.0	68.6	1
1	20200712	174	2	47.2197	14.7647	14.5	11.5	0.0	67.6	2
2	20200719	131	3	47.5822	19.2511	17.1	15.8	6.6	87.8	3

Figure 14: FactRaceWeather passed to the SSIS (latitude and longitude columns are amputated there)

6.3 DimDriver

The data in this table has been extracted from the GitHub data source. In the source, there were individual tables with drivers for each season, so we had to concatenate them, get rid of duplicates (two drivers had doubled rows with different data - deleted the incorrect ones), and amputate some columns that we didn't need. We also generated a unique ID for each driver.
At the end, after executing the whole Python script, we were left with this dataframe:

DriverID	Driver	Abbreviation	Country	Date of Birth	Place of Birth
001	Lewis Hamilton	HAM	United Kingdom	07/01/1985	Stevenage, England
002	Valtteri Bottas	BOT	Finland	28/08/1989	Nastola, Finland
003	Max Verstappen	VER	Netherlands	30/09/1997	Hasselt, Belgium
004	Sergio Perez	PER	Mexico	26/01/1990	Guadalajara, Mexico
005	Daniel Ricciardo	RIC	Australia	01/07/1989	Perth, Australia
006	Carlos Sainz	SAI	Spain	01/09/1994	Madrid, Spain
007	Alexander Albon	ALB	Thailand	23/03/1996	London, England
008	Charles Leclerc	LEC	Monaco	16/10/1997	Monte Carlo, Monaco
009	Lando Norris	NOR	United Kingdom	13/11/1999	Bristol, England
010	Pierre Gasly	GAS	France	07/02/1996	Rouen, France
011	Lance Stroll	STR	Canada	29/10/1998	Montreal, Canada
012	Esteban Ocon	OCO	France	17/09/1996	Evreux, Normandy

Figure 15: Drivers dataframe, later converted into csv and loaded into the data warehouse

6.4 FactRaceResult

This table was mostly populated with the use of .csv race results tables from the source on GitHub:

	Track	Position	No	Driver	Team	Starting Grid	Laps	Total Time/Gap/Retirement	Points	Fastest Lap	Year
0	Austria	1	77	Valtteri Bottas	Mercedes	1.0	71	1:30:55.739	25.0	No	2020
1	Austria	2	16	Charles Leclerc	Ferrari	7.0	71	+2.700	18.0	No	2020
2	Austria	3	4	Lando Norris	McLaren Renault	3.0	71	+5.491	16.0	Yes	2020

Figure 16: The initial data frame from GitHub

This data frame collected (almost) all race results for every driver in each race. The times, starting and finishing positions, drivers' names and their teams or how many classification points they got thanks to their results.

Transformations we needed to perform were:

1. Connecting with the other fact table through the DimDate table. As we encountered typo and naming problems again, we couldn't simply join the two tables by the track name and get the date. Fortunately, (after a manual check) it turned out that the both tabels were chronologically populated and the only obstacle we had to face was finding one race that didn't have all entries.

2. Thanks to the previous point, we were able to retain the circuit location as well. We decided that we want to keep it here as well so that we have a direct connection to the Circuits table - this might save users a dreadful multi-join when they just want to compare drivers' results with the tracks, not taking weather conditions into consideration.

3. Connecting to DimDriver table by unique driver ID - achieved by joining with the DimDriver table by driver's name - didn't come across any problems - retaining the IDs and deleting other driver columns.

4. Handle problems concerning incoherent data between 'Position' and 'Total Time [...]' columns.

Final data frame, obviously without final column names yet, looked as follows:

	Position	Team	Starting Grid	Total Time/Gap/Retirement	Points	Fastest Lap	DateID	DriverID	CircuitLocationID_x
0	1	Mercedes	1.0	1:30:55.739	25.0	No	20200705	2	174
1	2	Ferrari	7.0	+2.700	18.0	No	20200705	8	174
2	3	McLaren Renault	3.0	+5.491	16.0	Yes	20200705	9	174

Figure 17: Beginning of the data frame passed as a FactRaceResult to SSIS

6.5 DimDate

To populate this table we used the script from laboratories. (*LoadData_DimDate.sql*).

7 Potencial reports for users

Here we will present some interesting report ideas for future users of our data warehouse.

7.1 Driver Performance Analysis in Different Weather Conditions

Exemplary visualizations:

- Interactive charts showing the impact of weather conditions on final driver positions.
- Comparative charts for different seasons, teams, and drivers.

. Filtering options:

- By season (e.g., 2022, 2023)
- By driver or team (e.g., Lewis Hamilton, Ferrari)
- By track type (street vs. permanent circuit)
- By number of turns on track

Statistical summaries:

- Average final position based on rainfall
- Best and worst results for drivers in challenging weather

7.2 Season Overview and Championship Results

Exemplary visualizations:

- Progress charts for drivers and teams throughout a season.
- Season-to-season comparisons for drivers, teams, and circuits.

Filtering options:

- By season (e.g., 2022, 2023)
- By driver or team (e.g., Lewis Hamilton, Ferrari)

Statistical summaries:

- Average position
- Best and worst drivers performing in challenging weather
- Biggest performance gains and losses during a season.

7.3 Geographic Analysis of Circuits and Race Locations

Exemplary visualizations:

- Interactive maps showing circuit locations with details on turn count, circuit type etc.

Filtering options:

- By track type
- By season

Statistical summaries:

- the most challenging circuits in the history of Formula 1

8 Exemplary visualizatons in Power BI

We created 5 pages with data visualizations, one for each of these topics:

1. Constructor performace
2. Driver performance
3. Circuit information maps
4. Circuit weather maps
5. Performance in different weather conditions

Of course these visualizations are only exemplary and many more could be performed.

8.1 Constructor Performance

Here is our page displaying plots regarding consturctor performance. These plots are:

- Average starting grid for each constructor, which says how good they perform in qualifying (bar chart)
- Total points per constructor in all seasons(bar chart)
- Total points for each constructor in seasons, one after another (line chart)
- Total points over all seasons for each constructor (pie chart)

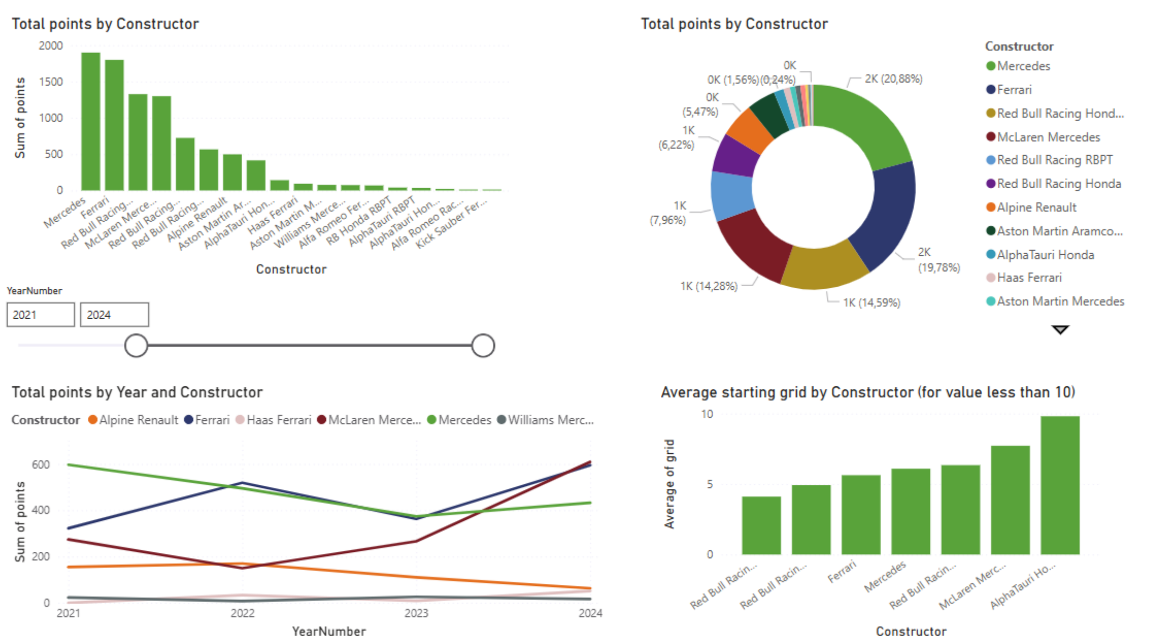


Figure 18: Constructors performance page

8.2 Driver Performance

Here is our page displaying plots regarding driver performance. These plots are:

- Total points per driver during the seasons for 6 selected drivers (line chart)
- Total points for each driver (pie chart)
- Total points over all seasons for each driver.

We also added two filtering options:

- By driver nationality (button slicer)
- By season range (slicer)

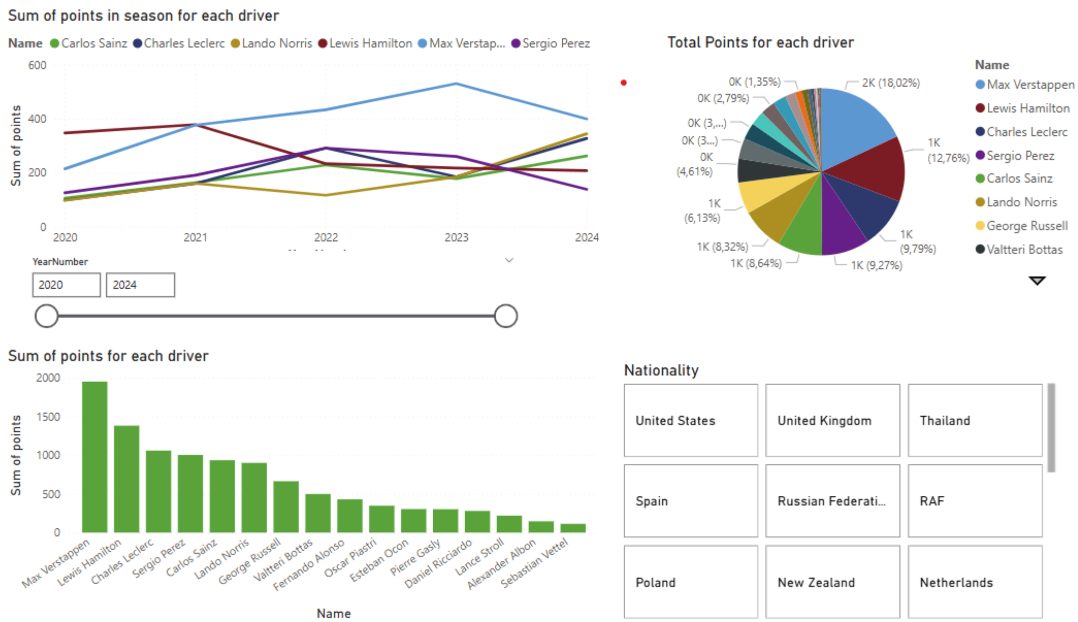


Figure 19: Drivers performance page - without filtering

Here is the page view with applied filtering:

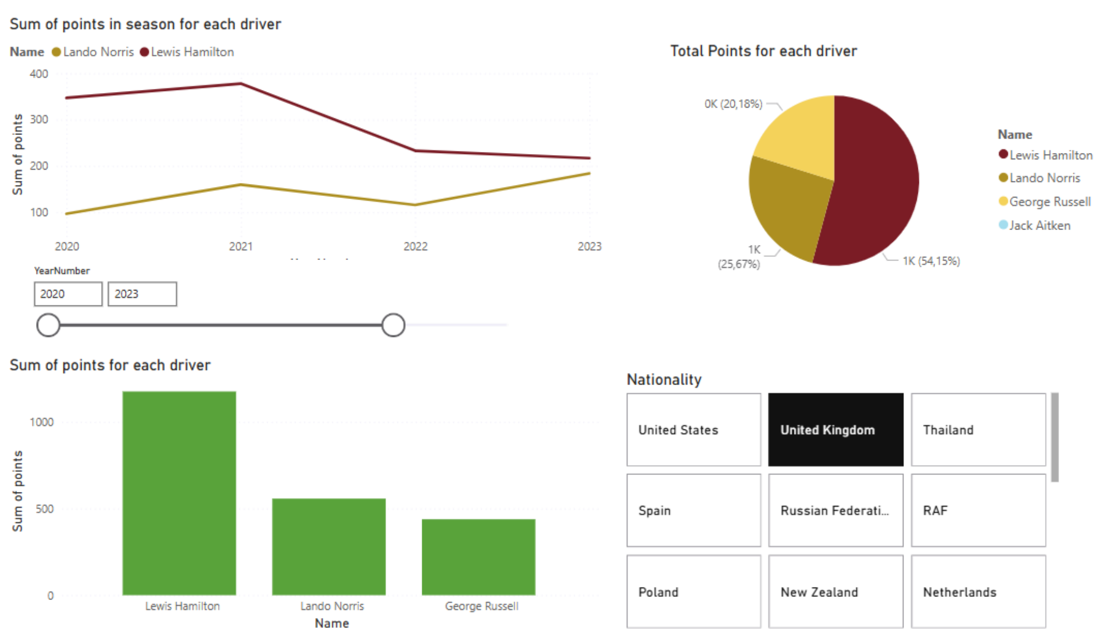


Figure 20: Drivers performance page - with filtering

8.3 Circuit Information maps

On this page we displayed some key information about the circuits:

- Circuit type - displayed on the first map and on the pie chart
- Number of Grand Prix held at this facility -displayed on the second map and the table

The table has been added to verify the values displayed on the map and to allow filtering.

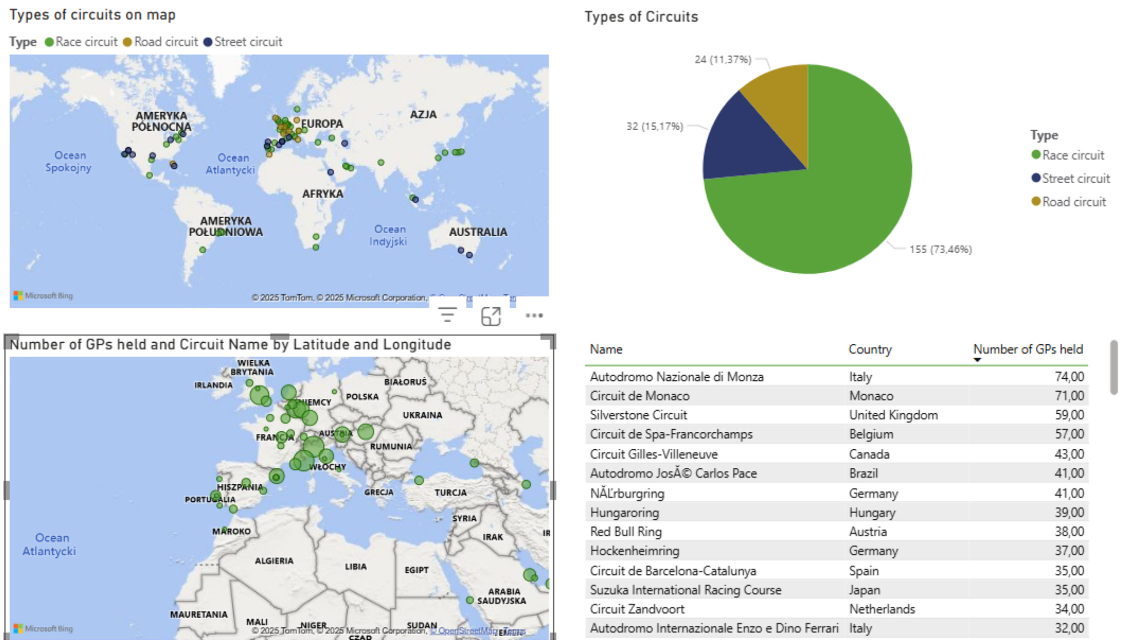


Figure 21: Maps with circuit infomation

8.4 Circuit Weather page

On this page we wanted to analyze the usual weather conditions on different circuits. To this this we prepared 4 visuzalisations:

- A distribution plot of the circuits, regarding average rain and average temperature, with 80th percentile annotated for each of those values (scatter plot)
- Average Temperature for circuits by Country (Tree map)
- Average Rain for Each Circuit on a map
- Average Rain for Each Circuit (bar chart)

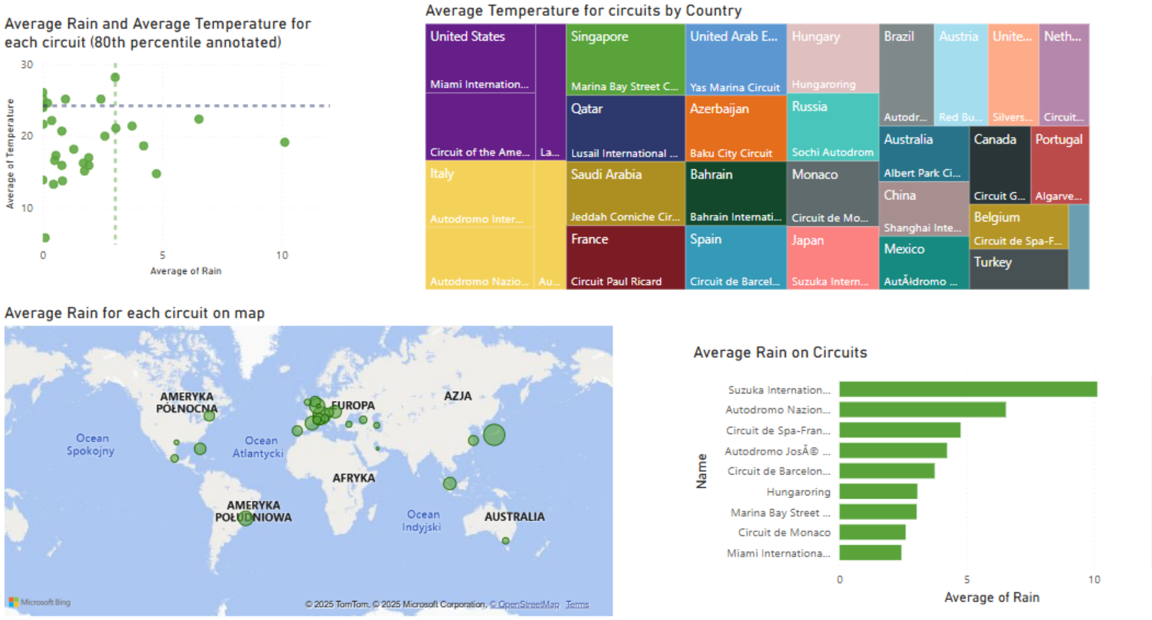


Figure 22: Circuit Weather page

8.5 Performance in different weather conditions

This page has been created, because comparing driver/constructor performance in different weather conditions is what inspired us to choose this project.
On this page we decided to show:

- Average points scores by selected drivers in different rain precipitation levels (line chart)
- Average points scores by selected drivers in different temperatures (line chart)
- The circuits with most combined DNFs, where DNF stand for did not finish, so the driver who DNFed either has an accident or retired from the race because of technical reasons (bar chart)
- Some card with basic weather aggregations

To create the DNF plot, we created a new measure in Power BI, which counted how many timer per race does the "DNF" value appear in the "time" column of FactRaceResult table.

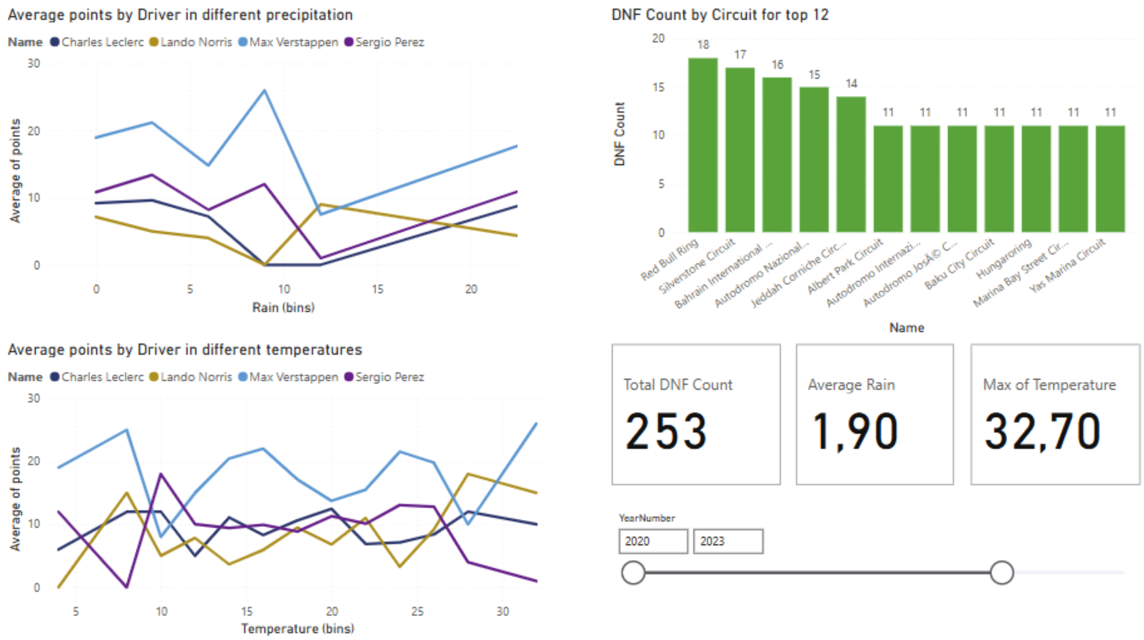


Figure 23: Performance in different weather conditions page

9 Data Warehouse testing

9.1 Data loading phase

Here, to prove that the data has been loaded to the data warehouse correctly, are screenshots from successful data loading in SSIS and screenshots of sql queries showing that the data is actually in the table for each table.

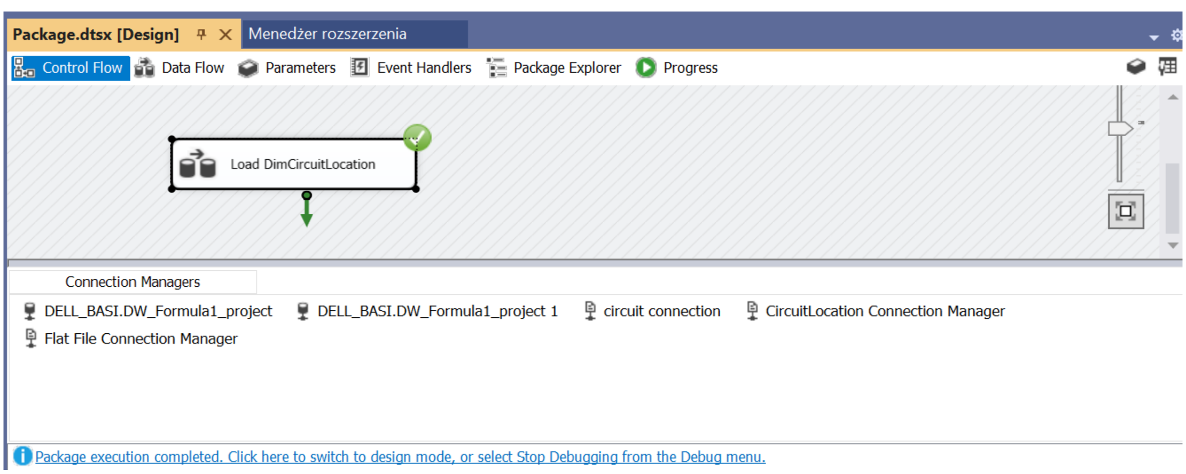


Figure 24: DimCircuitLocation table successful loading in SSIS

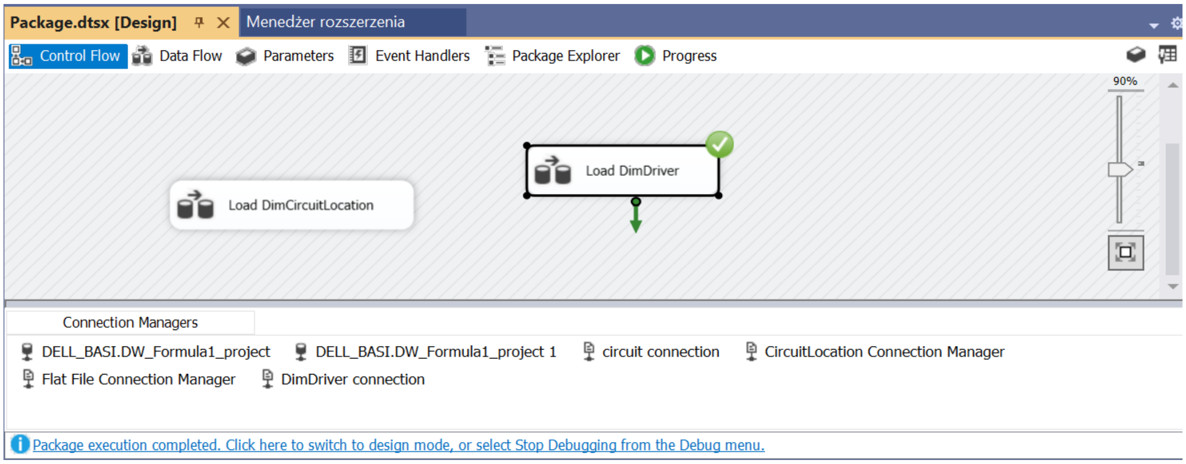


Figure 25: DimDriver table successful loading in SSIS

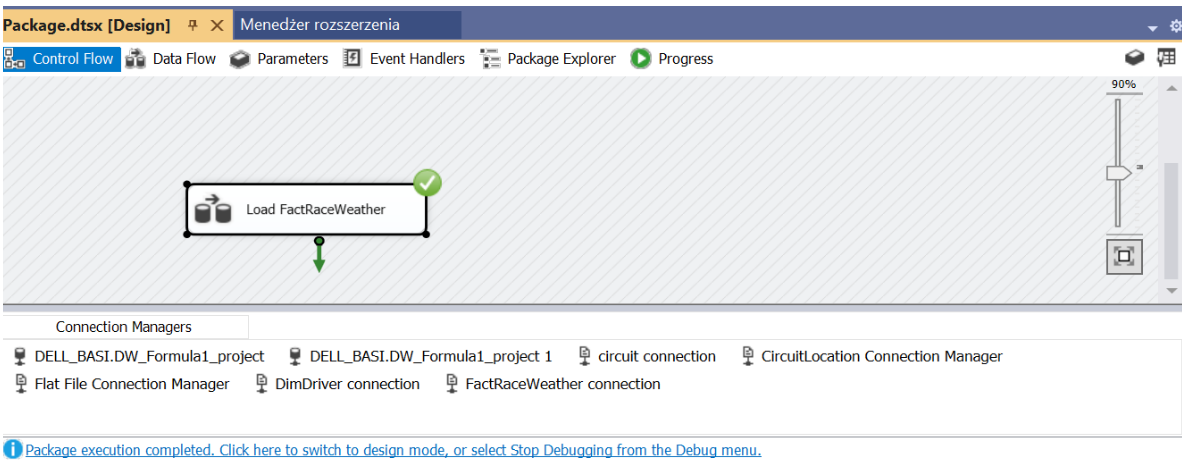


Figure 26: FactRaceWeather table successful loading in SSIS

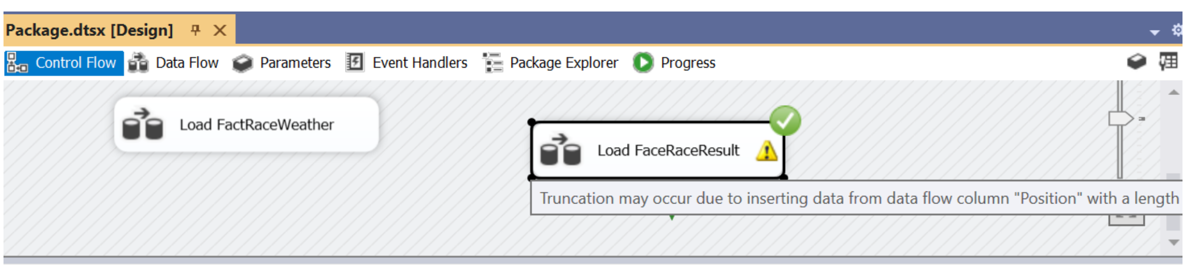
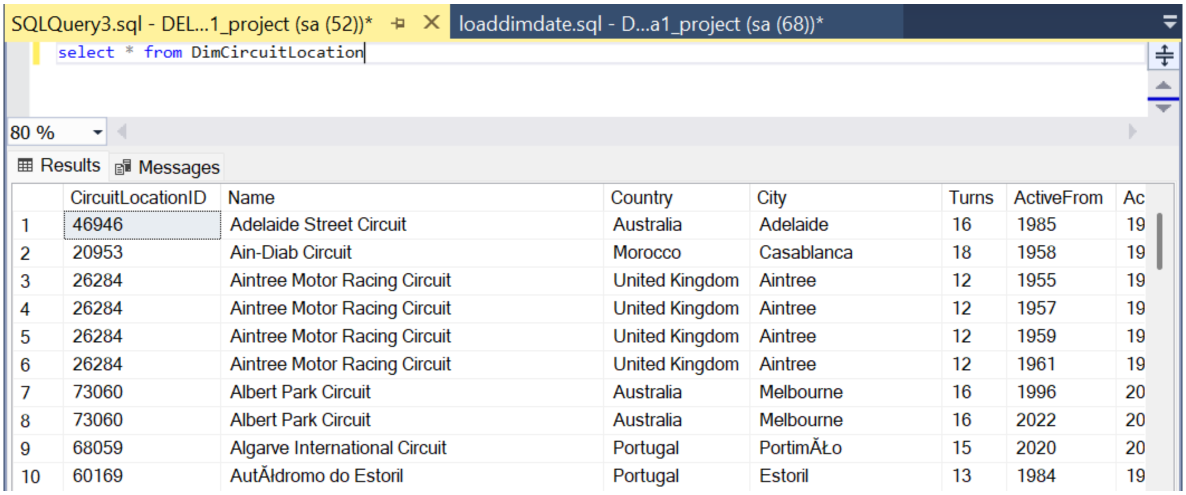
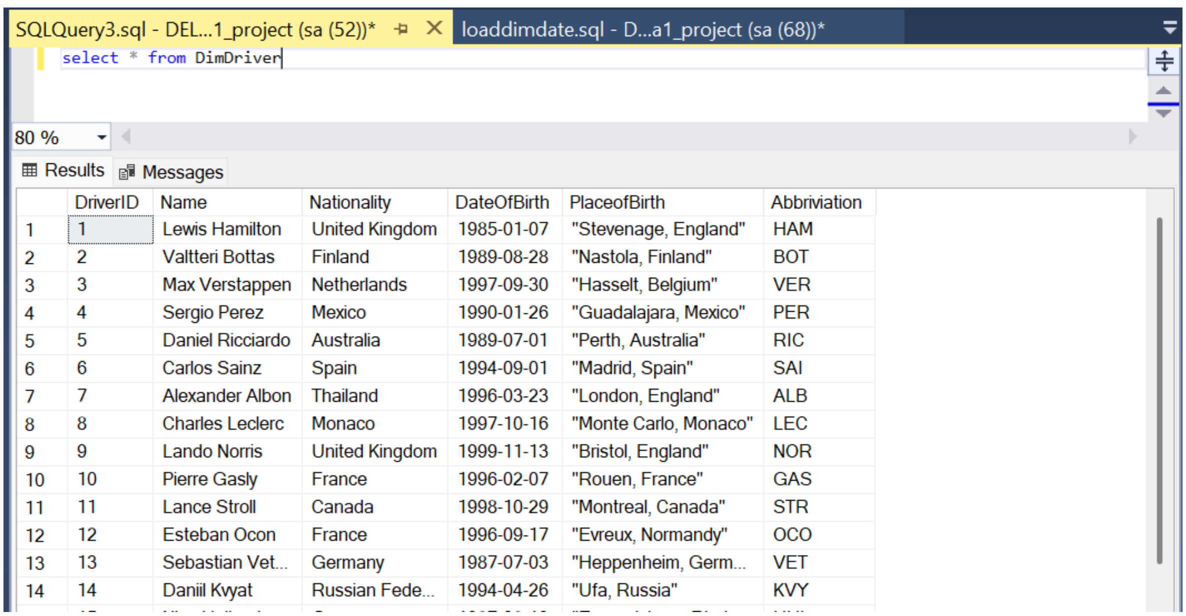


Figure 27: FactRaceResult table successful loading in SSIS



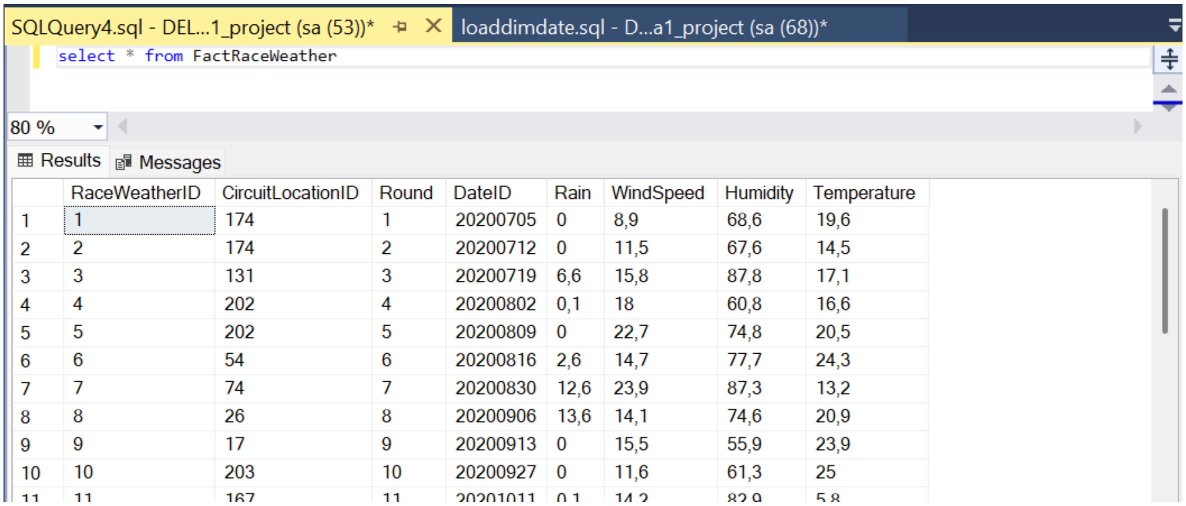
	CircuitLocationID	Name	Country	City	Turns	ActiveFrom	Ac
1	46946	Adelaide Street Circuit	Australia	Adelaide	16	1985	19
2	20953	Ain-Diab Circuit	Morocco	Casablanca	18	1958	19
3	26284	Aintree Motor Racing Circuit	United Kingdom	Aintree	12	1955	19
4	26284	Aintree Motor Racing Circuit	United Kingdom	Aintree	12	1957	19
5	26284	Aintree Motor Racing Circuit	United Kingdom	Aintree	12	1959	19
6	26284	Aintree Motor Racing Circuit	United Kingdom	Aintree	12	1961	19
7	73060	Albert Park Circuit	Australia	Melbourne	16	1996	20
8	73060	Albert Park Circuit	Australia	Melbourne	16	2022	20
9	68059	Algarve International Circuit	Portugal	PortimÃ£o	15	2020	20
10	60169	AutÃ³dromo do Estoril	Portugal	Estoril	13	1984	19

Figure 28: DimCircuitLocation query showing the successful data loading



	DriverID	Name	Nationality	DateOfBirth	PlaceofBirth	Abbriviation
1	1	Lewis Hamilton	United Kingdom	1985-01-07	"Stevenage, England"	HAM
2	2	Valtteri Bottas	Finland	1989-08-28	"Nastola, Finland"	BOT
3	3	Max Verstappen	Netherlands	1997-09-30	"Hasselt, Belgium"	VER
4	4	Sergio Perez	Mexico	1990-01-26	"Guadalajara, Mexico"	PER
5	5	Daniel Ricciardo	Australia	1989-07-01	"Perth, Australia"	RIC
6	6	Carlos Sainz	Spain	1994-09-01	"Madrid, Spain"	SAI
7	7	Alexander Albon	Thailand	1996-03-23	"London, England"	ALB
8	8	Charles Leclerc	Monaco	1997-10-16	"Monte Carlo, Monaco"	LEC
9	9	Lando Norris	United Kingdom	1999-11-13	"Bristol, England"	NOR
10	10	Pierre Gasly	France	1996-02-07	"Rouen, France"	GAS
11	11	Lance Stroll	Canada	1998-10-29	"Montreal, Canada"	STR
12	12	Esteban Ocon	France	1996-09-17	"Evreux, Normandy"	OCO
13	13	Sebastian Vet...	Germany	1987-07-03	"Heppenheim, Germ...	VET
14	14	Daniil Kvyat	Russian Fede...	1994-04-26	"Ufa, Russia"	KVY

Figure 29: DimDriver query showing the successful data loading



SQLQuery4.sql - DEL...1_project (sa (53))* loaddimdate.sql - D...a1_project (sa (68))*

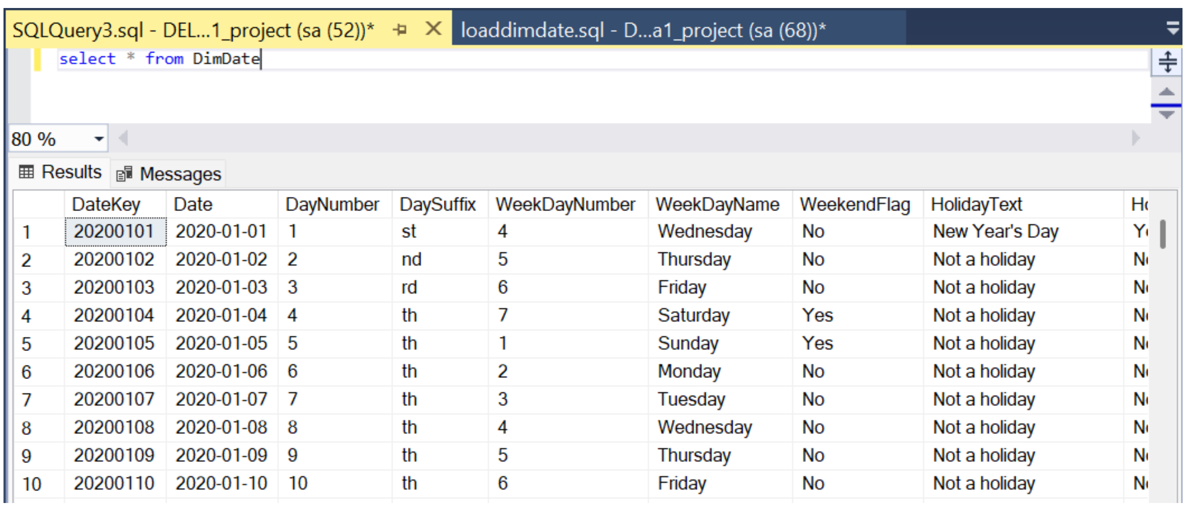
```
select * from FactRaceWeather
```

80 %

Results Messages

	RaceWeatherID	CircuitLocationID	Round	DateID	Rain	WindSpeed	Humidity	Temperature
1	1	174	1	20200705	0	8,9	68,6	19,6
2	2	174	2	20200712	0	11,5	67,6	14,5
3	3	131	3	20200719	6,6	15,8	87,8	17,1
4	4	202	4	20200802	0,1	18	60,8	16,6
5	5	202	5	20200809	0	22,7	74,8	20,5
6	6	54	6	20200816	2,6	14,7	77,7	24,3
7	7	74	7	20200830	12,6	23,9	87,3	13,2
8	8	26	8	20200906	13,6	14,1	74,6	20,9
9	9	17	9	20200913	0	15,5	55,9	23,9
10	10	203	10	20200927	0	11,6	61,3	25
11	11	167	11	20201011	0,1	14,2	82,9	5,8

Figure 30: FactRaceWeather query showing the successful data loading



SQLQuery3.sql - DEL...1_project (sa (52))* loaddimdate.sql - D...a1_project (sa (68))*

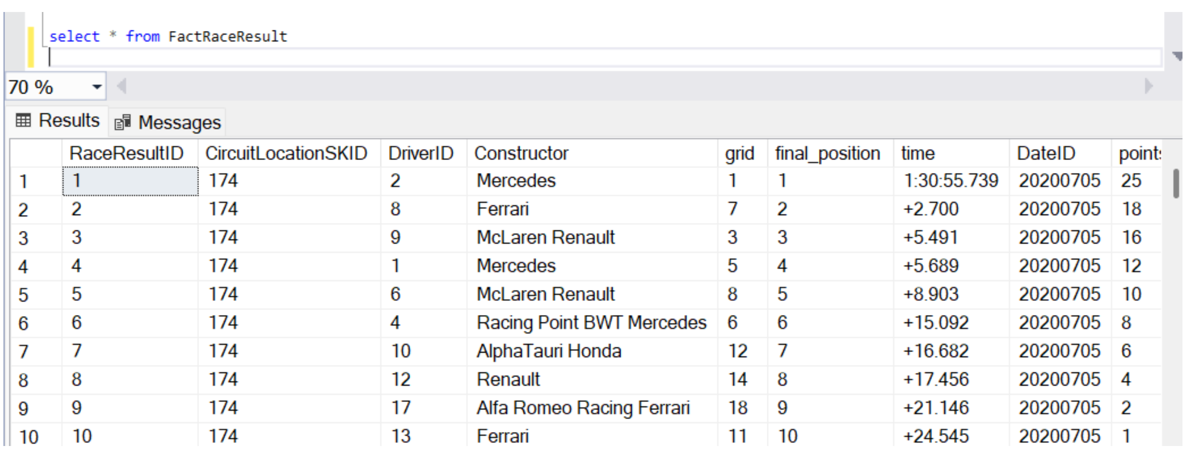
```
select * from DimDate
```

80 %

Results Messages

	DateKey	Date	DayNumber	DaySuffix	WeekDayNumber	WeekDayName	WeekendFlag	HolidayText	Ho
1	20200101	2020-01-01	1	st	4	Wednesday	No	New Year's Day	Y
2	20200102	2020-01-02	2	nd	5	Thursday	No	Not a holiday	N
3	20200103	2020-01-03	3	rd	6	Friday	No	Not a holiday	N
4	20200104	2020-01-04	4	th	7	Saturday	Yes	Not a holiday	N
5	20200105	2020-01-05	5	th	1	Sunday	Yes	Not a holiday	N
6	20200106	2020-01-06	6	th	2	Monday	No	Not a holiday	N
7	20200107	2020-01-07	7	th	3	Tuesday	No	Not a holiday	N
8	20200108	2020-01-08	8	th	4	Wednesday	No	Not a holiday	N
9	20200109	2020-01-09	9	th	5	Thursday	No	Not a holiday	N
10	20200110	2020-01-10	10	th	6	Friday	No	Not a holiday	N

Figure 31: DimDate query showing the successful data loading, using the sql query from laboratories



```
select * from FactRaceResult
```

70 %

Results Messages

	RaceResultID	CircuitLocationSKID	DriverID	Constructor	grid	final_position	time	DateID	point
1	1	174	2	Mercedes	1	1	1:30:55.739	20200705	25
2	2	174	8	Ferrari	7	2	+2.700	20200705	18
3	3	174	9	McLaren Renault	3	3	+5.491	20200705	16
4	4	174	1	Mercedes	5	4	+5.689	20200705	12
5	5	174	6	McLaren Renault	8	5	+8.903	20200705	10
6	6	174	4	Racing Point BWT Mercedes	6	6	+15.092	20200705	8
7	7	174	10	AlphaTauri Honda	12	7	+16.682	20200705	6
8	8	174	12	Renault	14	8	+17.456	20200705	4
9	9	174	17	Alfa Romeo Racing Ferrari	18	9	+21.146	20200705	2
10	10	174	13	Ferrari	11	10	+24.545	20200705	1

Figure 32: FactRaceResult query showing the successful data loading

9.2 Foreign keys

Additionality to make sure that all foreign keys are mapped correctly, we performed a very simple check on each relation:

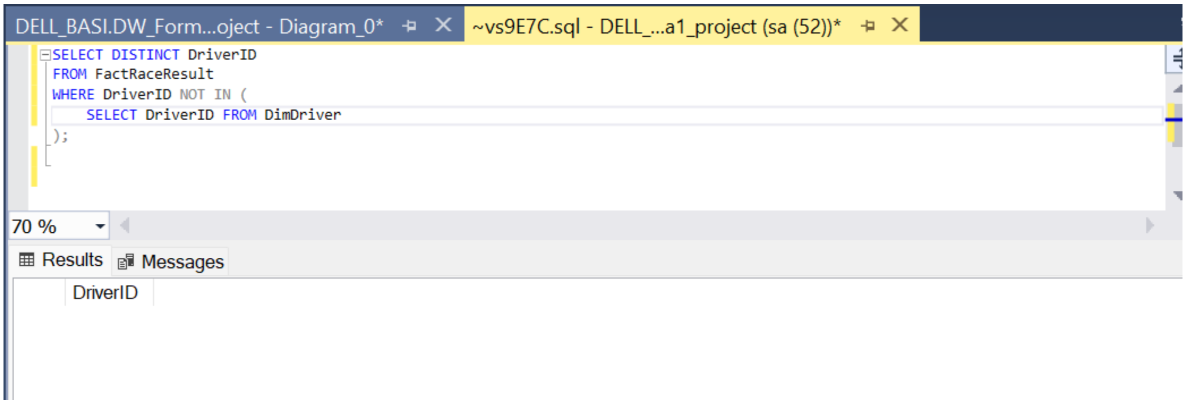


Figure 33: Foreign key validity check

In this validation, getting a blank answer is what we anticipated, since the query checks if in the fact table there are any values in foreign key columns that do not match the primary key in the table in the relation.
For every relation, we got a blank answer in this test, so we proceeded.

9.3 Working SCD2

We used SCD2 in the DimCircuitLocation table, since many circuits were only active for a short sequence of seasons, not throught the whole Formula 1 history. Also there are many cases where circuits kept disappearing and reappearing on the season calendar throughout the years (because of renovations, natural disasters etc.).
Thats why we decided to use SCD2 here. Here is an example of how it works.

130	10012	Hockenheimring	Germany	Hockenheim	16	2018	2019	False
131	59141	Hungaroring	Hungary	MogyorĂłd	14	1986	9999	True
132	90940	Indianapolis Motor Speedway	United States	Speedway	13	1950	1960	False
133	90940	Indianapolis Motor Speedway	United States	Speedway	13	2000	2007	False
134	59748	Intercity Istanbul Park	Turkey	Istanbul	14	2005	2011	False
135	59748	Intercity Istanbul Park	Turkey	Istanbul	14	2020	2021	False
136	58816	Jeddah Corniche Circuit	Saudi Arabia	Jeddah	27	2021	9999	True
137	95361	Korea International Circuit	South Korea	Yeongam	18	2010	2013	False
138	36587	Kyalami Grand Prix Circuit	South Africa	Midrand	13	1967	1980	False
139	36587	Kyalami Grand Prix Circuit	South Africa	Midrand	13	1982	1985	False
140	36587	Kyalami Grand Prix Circuit	South Africa	Midrand	13	1992	1993	False
141	77635	Las Vegas Strip Circuit	United States	Paradise	17	2023	9999	True

Figure 34: DimCircuitLocation SCD2 demonstration

Here we can observe that:

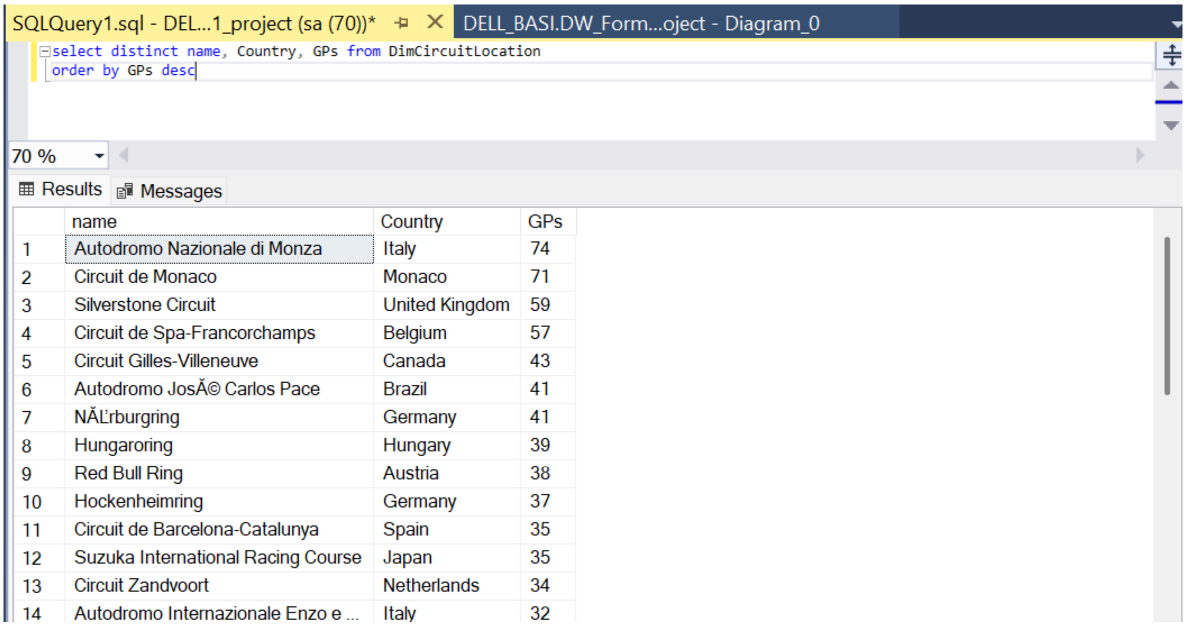
- in the last row we can see the Las Vegas GP, which has been added in 2023, and is still on the calendar, so that's why is has the ActiveTo value is set to the "end of the world year" (9999) and isActive flag set to True. Same for Hungaroring and Jeddah, which are also active circuits.
- for circuits that are not active anymore, the isActive flag is set to false
- for circuits that have appeared on the calendar a few times, we have 2 or more rows here. Each of them has its own SKID, but they share CircuitLocationID value.

9.4 Verifying the vizualizations

Here just to make sure that the data on the visualization is correct, we ran a few SQL queries in SQL Server and compared the results with what we got on the visualizations.

9.4.1 Circuit Information table

On the Circuit Information Page, we had a table that displayed each circuit, its country and number of Grand Prix held there. Lets check if in our data warehouse we will get the same result.



	name	Country	GPs
1	Autodromo Nazionale di Monza	Italy	74
2	Circuit de Monaco	Monaco	71
3	Silverstone Circuit	United Kingdom	59
4	Circuit de Spa-Francorchamps	Belgium	57
5	Circuit Gilles-Villeneuve	Canada	43
6	Autodromo JosĂ© Carlos Pace	Brazil	41
7	NĂĽrburgring	Germany	41
8	Hungaroring	Hungary	39
9	Red Bull Ring	Austria	38
10	Hockenheimring	Germany	37
11	Circuit de Barcelona-Catalunya	Spain	35
12	Suzuka International Racing Course	Japan	35
13	Circuit Zandvoort	Netherlands	34
14	Autodromo Internazionale Enzo e ...	Italy	32

Figure 35: Results in data warehouse

Name	Country	Number of GPs held
Autodromo Nazionale di Monza	Italy	74,00
Circuit de Monaco	Monaco	71,00
Silverstone Circuit	United Kingdom	59,00
Circuit de Spa-Francorchamps	Belgium	57,00
Circuit Gilles-Villeneuve	Canada	43,00
Autodromo JosĂ© Carlos Pace	Brazil	41,00
NĂĽrburgring	Germany	41,00
Hungaroring	Hungary	39,00
Red Bull Ring	Austria	38,00
Hockenheimring	Germany	37,00
Circuit de Barcelona-Catalunya	Spain	35,00
Suzuka International Racing Course	Japan	35,00
Circuit Zandvoort	Netherlands	34,00
Autodromo Internazionale Enzo e Dino Ferrari	Italy	32,00

Figure 36: Table in our visualization

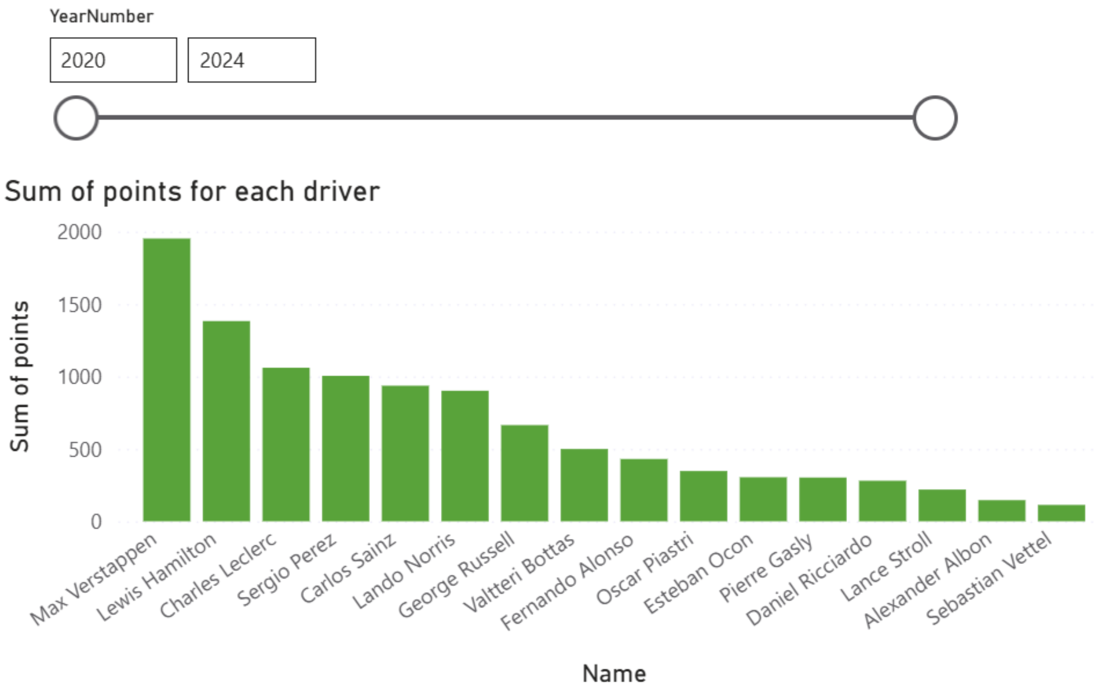
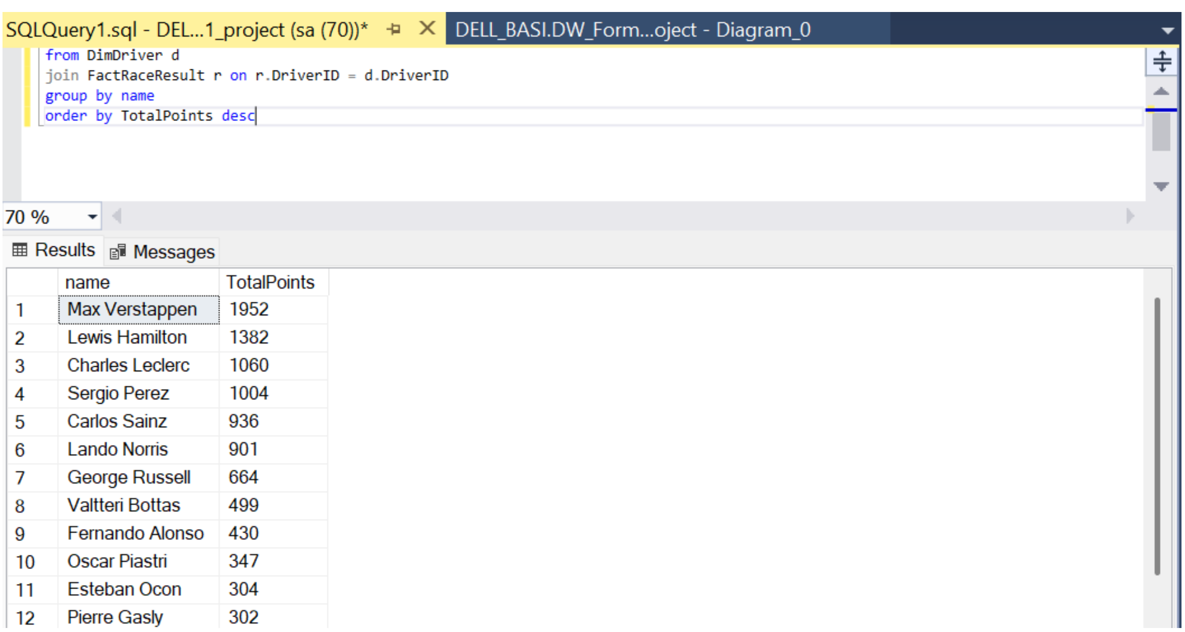


Figure 38: Drivers' total points bar chart from the Drivers Performance page

All the values here are the same, so this data passed the test.

9.4.2 Driver's total points

On the Drivers Performance page, we had a bar chart displying total points for each driver. Lets check if these results are also correct.


SQLQuery1.sql - DEL...1_project (sa (70))*　DELL_BASI.DW_Form...oject - Diagram_0

```sql
from DimDriver d
join FactRaceResult r on r.DriverID = d.DriverID
group by name
order by TotalPoints desc
```

70 %

Results　Messages

	name	TotalPoints
1	Max Verstappen	1952
2	Lewis Hamilton	1382
3	Charles Leclerc	1060
4	Sergio Perez	1004
5	Carlos Sainz	936
6	Lando Norris	901
7	George Russell	664
8	Valtteri Bottas	499
9	Fernando Alonso	430
10	Oscar Piastri	347
11	Esteban Ocon	304
12	Pierre Gasly	302

Figure 37: Drivers' total points query

Here we can observe that these values probably match, but just to make sure they are fully correct, we generated a table in Power BI to compare it to our result from the sql query.

Name	Sum of points
Max Verstappen	1952
Lewis Hamilton	1382
Charles Leclerc	1060
Sergio Perez	1004
Carlos Sainz	936
Lando Norris	901
George Russell	664
Valtteri Bottas	499
Fernando Alonso	430
Oscar Piastri	347
Esteban Ocon	304
Pierre Gasly	302
Daniel Ricciardo	280

Figure 39: Drivers' total points check

Here we can fully compare the tables and say that the data fully matches.

10 Github

The project was developed with using a GitHub repository. All python code, used for data transformation can be found there.
https://github.com/michalwietecki/f1-weather-dwh

11 Division of work

- project idea and business cases - both
- data search and exploration - both
- data warehouse structure (diagram and decriptions) - Basia
- ETL plan - both
- extracting and tranforming the data (python) - Michał
- loading the data into the data warehouse (SSIS) - Basia
- exemplary visualizations in Power BI and their testing - Basia