Master in Artificial Intelligence
Academic Year 2025-2026

October 2025

Data Engineering (P4251102)

Dimensional modelling and ETL data processing: Formula 1

Yago Estévez Figueiras[1],
Andrea Real Blanco[2],
Francisco Manuel Vázquez Fernández[3]

Abstract. The objective of this lab practice is to design and implement a dimensional model to support the analysis of Formula 1 (F1) performance. The data is organized using three star schemas encompassing Qualifying, Pit Stops, and Race Results. An ETL pipeline is developed and executed to extract, transform, and load the source data into the data warehouse.

Contents

[1]yago.estevez.figueiras@rai.usc.es
[2]andrea.real@rai.usc.es
[3]franciscomanuel.vazquez.fernandez@rai.usc.gal

1 Software and Requirements

To reproduce the pipeline and populate the data warehouse, the following software and steps are required:

1. Relational Database: MySQL and MySQL Workbench are used as the DBMS.
2. Run the provided `create_database_f1.sql` script to generate the database schemas.
3. Install the required Python packages (Recommended Python version 3.x):

   ```
   pip install pandas numpy sqlalchemy pymysql matplotlib
   ```

4. Run `pipeline.ipynb` to execute the ETL process and populate the MySQL database. The database connection parameters and data directory path are managed through the `CONFIG` dictionary and `DATA_DIR` variable in the Python code.

2 Data Analysis and Modelling

2.1 Analytical Objectives and Justification

Based on the provided data, we develop a model designed to support the analysis of drivers' and constructors' performance across seasons and circuits. With this purpose, we track the performance in three different key areas: qualifying times, pit stop efficiency, and race results, each with its own star schema.

Schema 1. Qualifying star schema (`fact_qualy`). The model enables answering questions such as:

- Which constructor shows the greatest relative q_3 speed advantage when racing at circuits located above 50 meters of altitude?
- How has the average difference between a driver's q_1 and q_3 times changed across years?
- Which team/constructor has made its way into q_3 the most times?

Schema 2. Pit stops star schema (`fact_pit`). The model allows answering questions such as:

- What is the average and standard deviation of pit stop time for each constructor in a certain season?
- At which circuits and lap numbers does a constructor average the fastest pit stop time in a certain season?
- How does a specific constructor's average pit stop time vary based on the lap number?

Schema 3. Race Results Star Schema (`fact_results`). The model allows answering questions such as:

- What is the average net change in position for drivers who start outside the top ten?
- What is the circuit with the most positions exchanged in a certain season?
- Which driver won more positions for a given race?

It should be noted that the analytical questions serve as examples of the type of queries the proposed database models can answer. They are intended for demonstration purposes, and do not correspond exactly to those visualized in the accompanying Tableau dashboard.

2.2 Dimensional Model Diagram

As mentioned earlier, we developed three star schemas, each with its own fact table and four dimensions shared among the three schemas.

- **Dimension Tables**

  - `dim_driver`: As our primary key (PK) ("driver_id" as a varchar(20)), we used the driver ref name from the database, as it is unique for each driver. This dimension stores the information about the name, surname, nationality, and the date of birth for each driver.
  - `dim_constructor`: As our PK ("constructor_id" as a varchar(25), we used the constructor ref name from the database, as it is unique for each team. This dimension stores the information about the name and the nationality for each team.
  - `dim_circuit`: As our PK ("circuit_id" as a varchar(20)), we also used the circuit ref name from the database, as it is unique for each circuit.
  - `dim_race`: As our PK ("race_id" as a varchar(10)), we used the date that a race took place, with the format "YYYYMMDD" because it is unique for each race. This dimension stores the information about the date that race took place as a date type, and the round of the season that race took place.
- **Fact Tables** All the fact_tables have an int auto_increment as PK and the foreign keys (FK) for establishing the relations between dimensions as the id in that dimension with the same format.

  - `fact_qualy`: We have our PK "qualy_id", all the FKs and the q1, q2 and q3 times in milliseconds as type "INT".

  - `fact_pit`: We have our PK "pit_id", all the FKs, and the information about the number of that stop "stop_number", the lap that stop took place "lap_number," and the time in pits

  - `fact_result`: We have our PK "result_id", all the FKs, the information about the starting and finish position for each driver in each race, "start_pos" and "finish_position", and the points scored in that race "points".

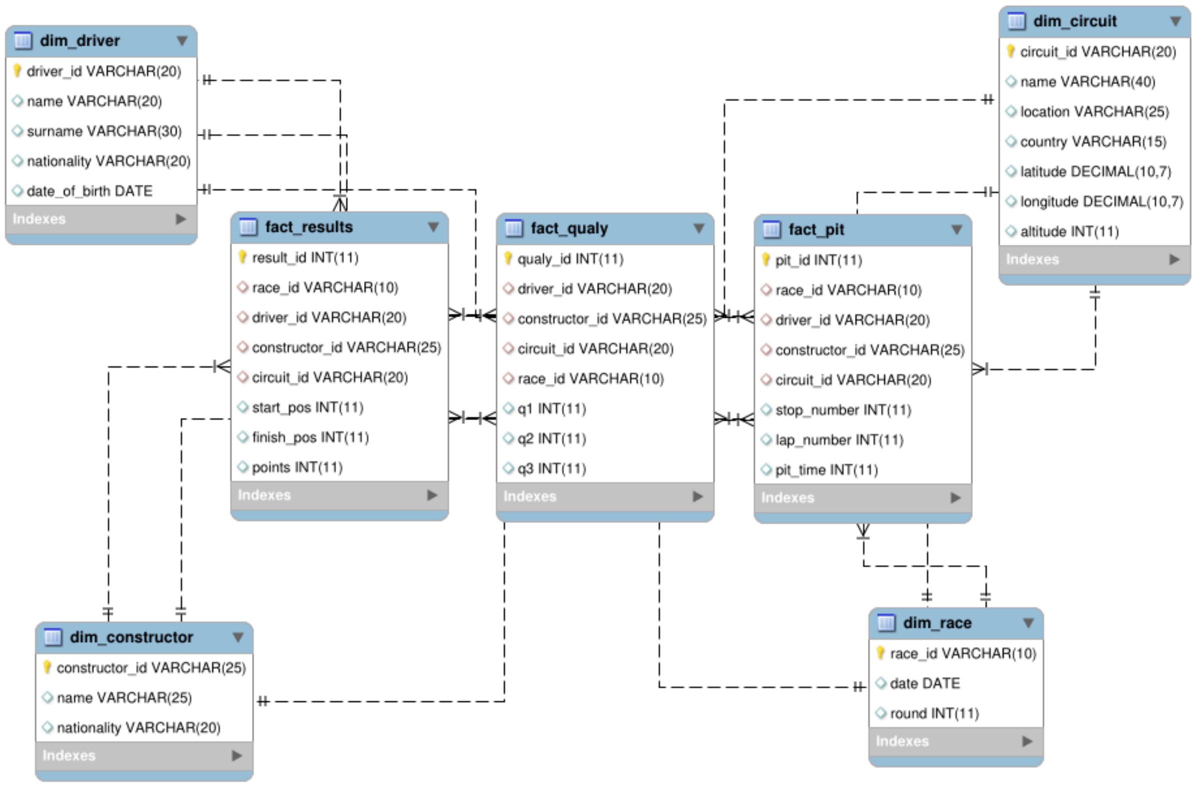


Figure 1: Dimensional star schema for the analysis of F1 performance data warehouse.

2.3 Key Data Analysis

In this section, we will introduce some concepts on how we should interpret the data from our database, and if it is coherent or if it has some errors.

First of all, the qualifying times, we find that 47% of the data does not have q2 values, and 68% for q3 values. If we are used to the nowadays qualifying system, these values seem normal, in each qualifying session 5 drivers are eliminated, and therefore if they have not classified at the q_1 they would not have a q_2 and q_3 time. However, we have historical data that goes from 2021 until the 1994 and if we check the numbers they do not match, the key factor here is that the current qualifying system was born in 2006, before this other qualifying methods were applied, and the times were stored as q1 values and zeros in q2 and q3, with an exception in 2005 when the FIA (Fédération Internationale de l'Automobile) implemented a two phased qualifying, so in this exceptional case we have in our database q1 and q2 values. This is important because if we want to make an analysis prior to 2006 we have to take this into account.

Moving to the pit times, our data goes from 2022 to 2011. We also find abnormal data in some races with high pit times. The first thought is that there was an error and the data should be discarded, but that would be a mistake, our database stores the values of the times since the car enters the pit lane until it exits the pit lane. This is different from the time it takes to change tires and may vary greatly from one circuit to another due to its design. So in this context, we can still find abnormal times, in this case related to race incidents, such as the 2021 Belgium GP known as the shortest F1 race in history. Due to the weather complications it was impossible to drive and many red flags were deployed, this meant that all cars must go into the pit lane and wait until the race restart, that explains the over 30 min times that we see in our database, and in general if a race flag is deployed and the race is stopped, it will be registered as a high pit time. In short, statistically anomalous data in this case contains important information about the events that occurred in the race and should not be discarded.

For the last model, we used the data from the results of each race, these results go from 2022 to 1950. The first thing we had to do was to get rid of the Null values stored as strings "N" and change them into "0" values, this makes sense in the context that a driver did not make it to the end of the race. The other thing that we shall take into account is that the points system also changes over time, with the biggest change in 2010 when the 1st place takes 25 points instead of the 10 points in 2009 and even less as we go further into the past. Taking this into account, we can still find abnormal values in some races. In some races the winner is given half of the points, this is no error and it is related to a rule that says that if the race finishes without reaching 75% of the programmed distance, the winners receive only half the points. In some seasons, an extra point is given to the driver that makes the fastest lap and finishes in the top ten. There was one exceptional case in 2014 at the Abu Dabi GP where the top ten would be given double points.

Another problem we found was in the circuits dimension, where the altitude of two circuits was stored as null. We replaced that with the real values, but replacing them with 0 would also be correct, as those circuits are near sea level.

So our database is consistent with the historical context and rules of each season, and **no data should be eliminated from the selection we made**.

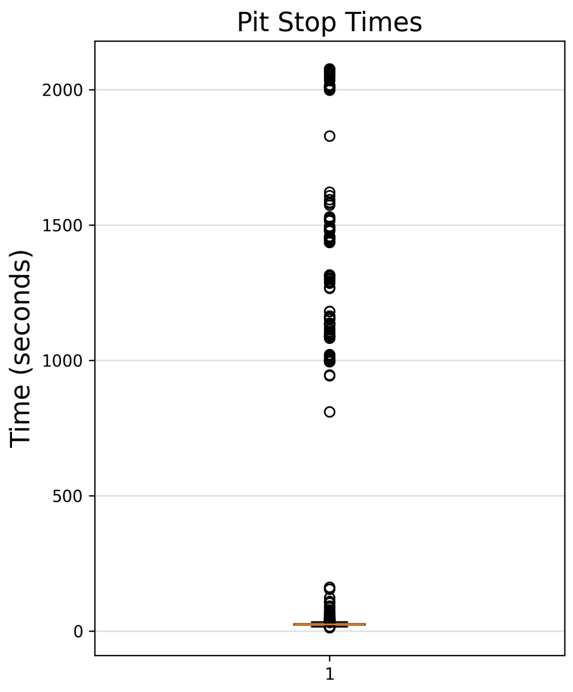


Figure 2: Distribution of pit stop times across all drivers. Outliers correspond to unusually long or short pit stops.

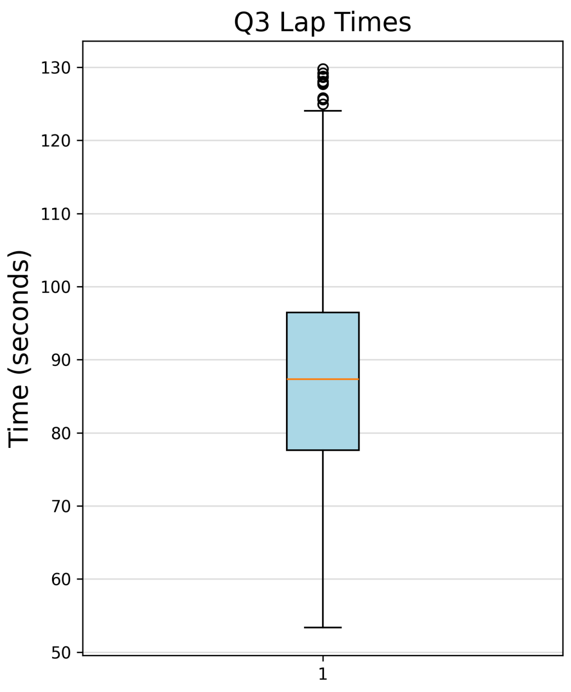


Figure 3: Distribution of q_3 lap times for all drivers. Outliers beyond the whiskers represent unusually slow or fast laps.

3 Data Pipeline (ETL)

The data pipeline is implemented using Python to perform extraction, transformation, and loading into the data warehouse.

3.1 Extraction

The raw data is extracted from the CSV files of the Kaggle dataset Formula 1 Pit Stops Analysis using the `pandas` library. During this initial load, data validation checks were executed to ensure source integrity prior to cleaning.

3.2 Transformation

The main operations carried out during the transformation phase can be classified into three areas:

- **Dimensional key creation**: All dimension tables are created by selecting core attributes and renaming columns for consistency with the designed database schema (e.g., 'driverRef' to 'driver_id'). The 'race_id' is derived from the date string to serve as a primary key.
- **Metric standardization**: The `q_to_ms` helper function is used to convert all qualifying time strings (q_1, q_2, q_3) into a standard integer unit (milliseconds), to make them ready for mathematical operations and analytical queries.
- **Fact table joins and null handling**: Fact DataFrames are constructed through a sequence of left merges to link all four-dimensional keys to the metrics. The `fact_results_df` function explicitly cleans the 'N' missing values in points and positions by replacing them with 0 and casting the columns to INT. In `dim_circuit_df`, missing values in the altitude column are manually added.

3.3 Load

The data is loaded into MySQL using the `.to_sql()` method with `SQLAlchemy`. The loading process is executed modularly to ensure that the pipeline for each star schema can be run independently.

- **Shared dimension load**: All four dimension tables are created and loaded first using their respective `load_dim_...()` functions. The explicit dtype mapping in each load function ensures the correct `VARCHAR` sizes and `DATE` formats are enforced in the database schema.
- **Single fact load**: Three separate functions (`load_fact_qualy()`, `load_fact_pit()`, `load_fact_results()`) are defined, with each function handling the loading of a single fact table (`fact_qualy`, `fact_pit`, `fact_results`) and its associated metrics.

```python
def load_fact_qualy(fact_qualy, engine):
    table_name = 'fact_qualy'
    try:
        fact_qualy.to_sql(
            table_name,
            con=engine,
            if_exists='append',
            index=False,
            dtype={
                "driver_id": VARCHAR(20),
                "constructor_id": VARCHAR(25),
                "race_id": VARCHAR(10),
                "circuit_id": VARCHAR(20),
                "q1": INT,
                "q2": INT,
                "q3": INT
        })
        print(f"Table '{table_name}' loaded successfully.")
    except Exception as e:
        print(f"Error loading table '{table_name}': {e}")
    return
```

Listing 1: Example code for loading a fact table into MySQL using pandas and SQLAlchemy.

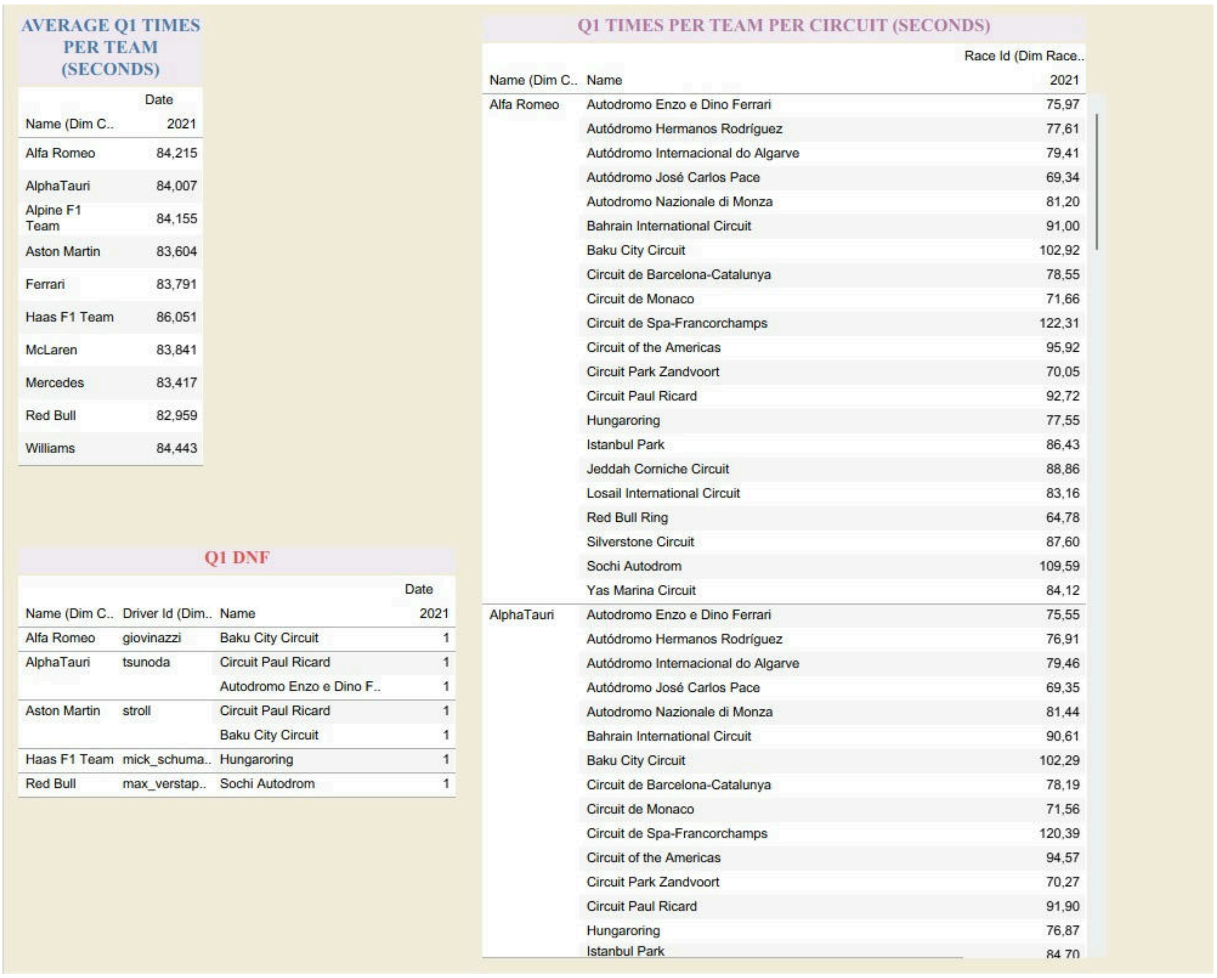
4 Data Analysis (Tableau)

First of all, in order to open the files, when Tableau opens a .twb file for the first time, you encounter that the design of the visualizations is there, but not the data: Tableau will give an error message like "data source not found", the easy way to solve this is by editing the connection and navigating to the .csv files required. It should be enough to select the specific fact table file for each tableau file. The other way is to establish the connection via the Tableau MySQL drivers with your local database credentials.

In order to make quick and visual analysis from our data, we developed some dashboards for each star schema, many of which focused on the 2021 season due to its relevance in the sport from an F1 fan perspective. Let's explain each one:

- If we open the "QUALY.twb" file, we can see two dashboards. The first one focuses on the q_1 times in the 2021 season, showing the average q_1 times per team, the amount of DNF per driver in q_1, and the mean q_1 times per team in each race. The second one shows the average q_3 times per driver from 2006 to 2021 and the amount of times a driver has made its way into q_3.

- If we open the “PITS.twb” file, we can see a dashboard with the information about the pit times per circuit and for each team in the 2021 season.
- If we open the “RESULTS.dwb” file, we can also see two dashboards. In the first one, we can see the total points per driver and the points breakdown for each race in the 2021 season. In the other dashboard, we can see all the drivers that win a race and the amount of wins per driver from 1955 until 2022.



AVERAGE Q1 TIMES PER TEAM (SECONDS)

	Date
Name (Dim C..	2021
Alfa Romeo	84,215
AlphaTauri	84,007
Alpine F1 Team	84,155
Aston Martin	83,604
Ferrari	83,791
Haas F1 Team	86,051
McLaren	83,841
Mercedes	83,417
Red Bull	82,959
Williams	84,443

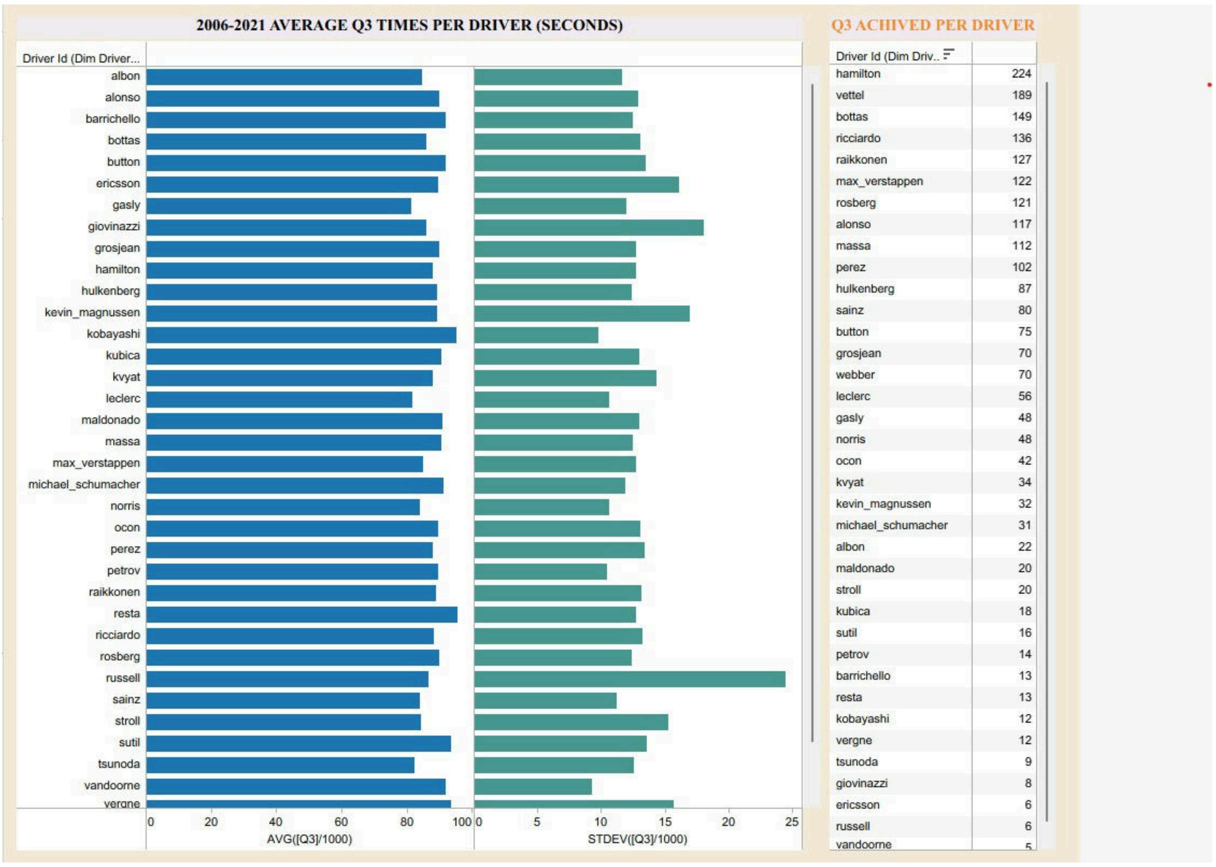
Q1 DNF

			Date
Name (Dim C..	Driver Id (Dim..	Name	2021
Alfa Romeo	giovinazzi	Baku City Circuit	1
AlphaTauri	tsunoda	Circuit Paul Ricard	1
		Autodromo Enzo e Dino F..	1
Aston Martin	stroll	Circuit Paul Ricard	1
		Baku City Circuit	1
Haas F1 Team	mick_schuma..	Hungaroring	1
Red Bull	max_verstap..	Sochi Autodrom	1

Q1 TIMES PER TEAM PER CIRCUIT (SECONDS)

		Race Id (Dim Race..
Name (Dim C..	Name	2021
Alfa Romeo	Autodromo Enzo e Dino Ferrari	75,97
	Autódromo Hermanos Rodríguez	77,61
	Autódromo Internacional do Algarve	79,41
	Autódromo José Carlos Pace	69,34
	Autodromo Nazionale di Monza	81,20
	Bahrain International Circuit	91,00
	Baku City Circuit	102,92
	Circuit de Barcelona-Catalunya	78,55
	Circuit de Monaco	71,66
	Circuit de Spa-Francorchamps	122,31
	Circuit of the Americas	95,92
	Circuit Park Zandvoort	70,05
	Circuit Paul Ricard	92,72
	Hungaroring	77,55
	Istanbul Park	86,43
	Jeddah Corniche Circuit	88,86
	Losail International Circuit	83,16
	Red Bull Ring	64,78
	Silverstone Circuit	87,60
	Sochi Autodrom	109,59
	Yas Marina Circuit	84,12
AlphaTauri	Autodromo Enzo e Dino Ferrari	75,55
	Autódromo Hermanos Rodríguez	76,91
	Autódromo Internacional do Algarve	79,46
	Autódromo José Carlos Pace	69,35
	Autodromo Nazionale di Monza	81,44
	Bahrain International Circuit	90,61
	Baku City Circuit	102,29
	Circuit de Barcelona-Catalunya	78,19
	Circuit de Monaco	71,56
	Circuit de Spa-Francorchamps	120,39
	Circuit of the Americas	94,57
	Circuit Park Zandvoort	70,27
	Circuit Paul Ricard	91,90
	Hungaroring	76,87
	Istanbul Park	84,70

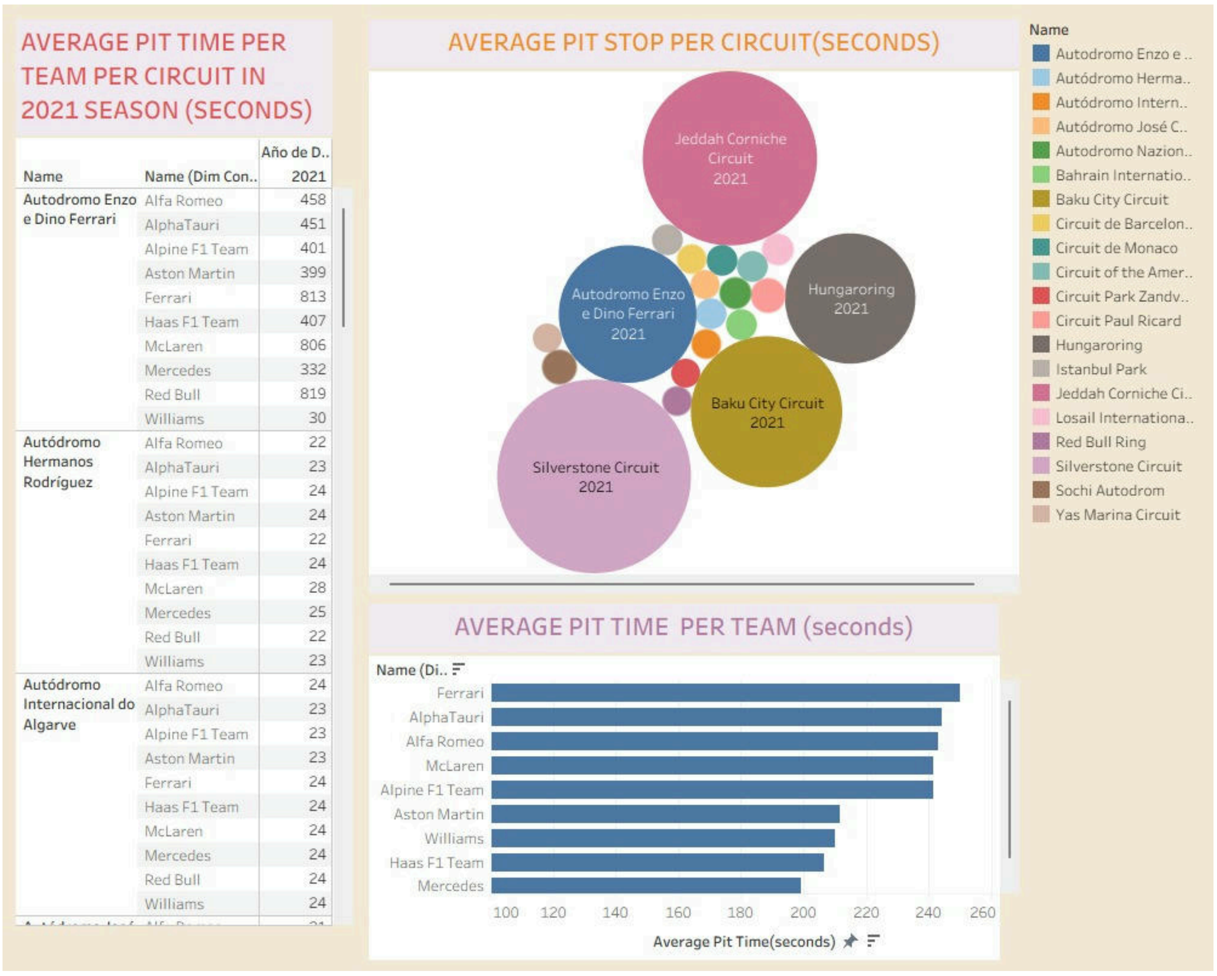
Figure 4: From “QUALY.twb”.



Q3 ACHIVED PER DRIVER

Driver Id (Dim Driv..	
hamilton	224
vettel	189
bottas	149
ricciardo	136
raikkonen	127
max_verstappen	122
rosberg	121
alonso	117
massa	112
perez	102
hulkenberg	87
sainz	80
button	75
grosjean	70
webber	70
leclerc	56
gasly	48
norris	48
ocon	42
kvyat	34
kevin_magnussen	32
michael_schumacher	31
albon	22
maldonado	20
stroll	20
kubica	18
sutil	16
petrov	14
barrichello	13
resta	13
kobayashi	12
vergne	12
tsunoda	9
giovinazzi	8
ericsson	6
russell	6
vandoorne	5

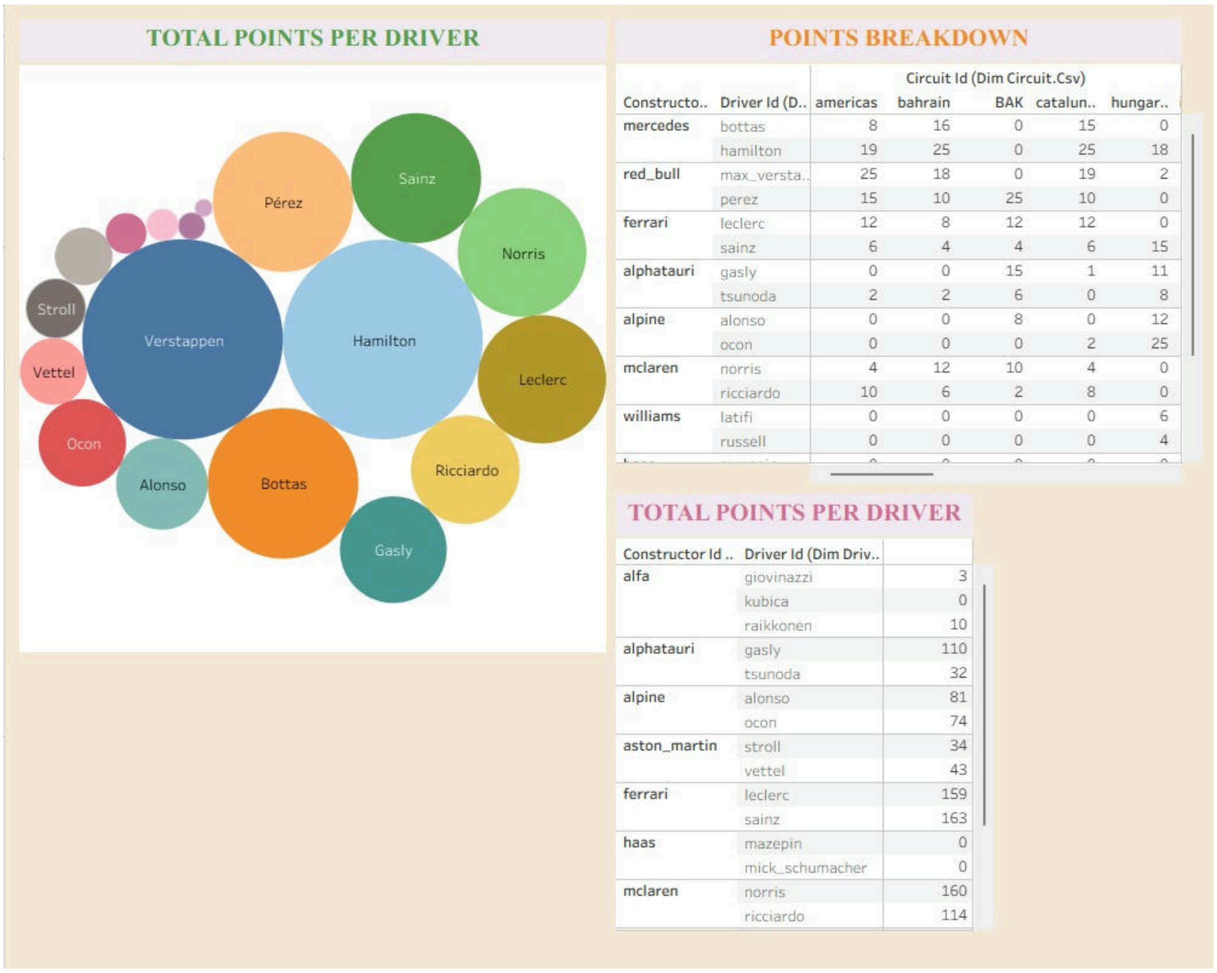
Figure 5: From “QUALY.twb”.



AVERAGE PIT TIME PER TEAM PER CIRCUIT IN 2021 SEASON (SECONDS)

Name	Name (Dim Con..	Año de D.. 2021
Autodromo Enzo e Dino Ferrari	Alfa Romeo	458
	AlphaTauri	451
	Alpine F1 Team	401
	Aston Martin	399
	Ferrari	813
	Haas F1 Team	407
	McLaren	806
	Mercedes	332
	Red Bull	819
	Williams	30
Autódromo Hermanos Rodríguez	Alfa Romeo	22
	AlphaTauri	23
	Alpine F1 Team	24
	Aston Martin	24
	Ferrari	22
	Haas F1 Team	24
	McLaren	28
	Mercedes	25
	Red Bull	22
	Williams	23
Autódromo Internacional do Algarve	Alfa Romeo	24
	AlphaTauri	23
	Alpine F1 Team	23
	Aston Martin	23
	Ferrari	24
	Haas F1 Team	24
	McLaren	24
	Mercedes	24
	Red Bull	24
	Williams	24

Figure 6: From "PITS.twb".



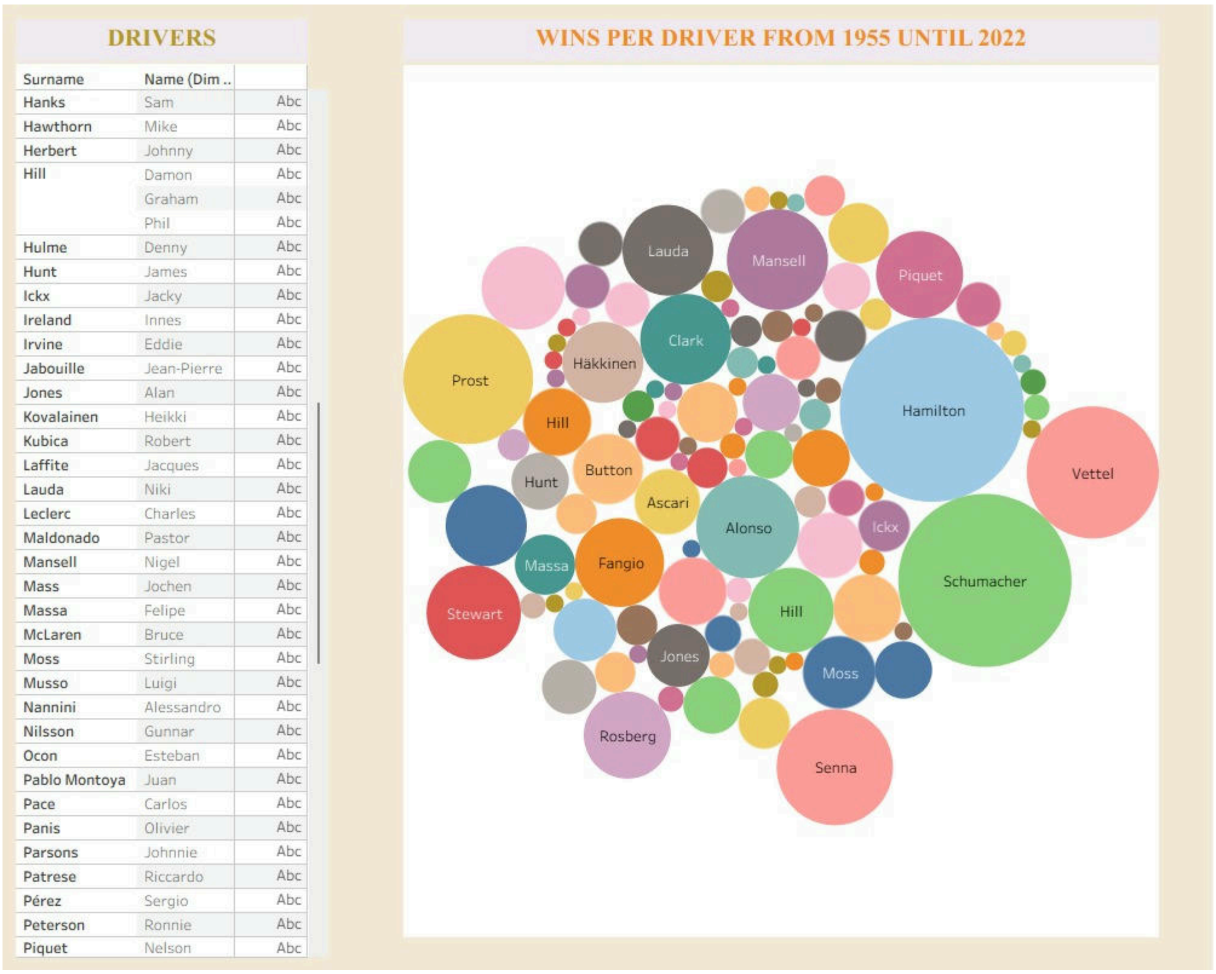
POINTS BREAKDOWN

Constructo..	Driver Id (D..	Circuit Id (Dim Circuit.Csv) americas	bahrain	BAK	catalun..	hungar..
mercedes	bottas	8	16	0	15	0
	hamilton	19	25	0	25	18
red_bull	max_versta..	25	18	0	19	2
	perez	15	10	25	10	0
ferrari	leclerc	12	8	12	12	0
	sainz	6	4	4	6	15
alphatauri	gasly	0	0	15	1	11
	tsunoda	2	2	6	0	8
alpine	alonso	0	0	8	0	12
	ocon	0	0	0	2	25
mclaren	norris	4	12	10	4	0
	ricciardo	10	6	2	8	0
williams	latifi	0	0	0	0	6
	russell	0	0	0	0	4

TOTAL POINTS PER DRIVER

Constructor Id ..	Driver Id (Dim Driv..	
alfa	giovinazzi	3
	kubica	0
	raikkonen	10
alphatauri	gasly	110
	tsunoda	32
alpine	alonso	81
	ocon	74
aston_martin	stroll	34
	vettel	43
ferrari	leclerc	159
	sainz	163
haas	mazepin	0
	mick_schumacher	0
mclaren	norris	160
	ricciardo	114

Figure 7: From “RESULTS.twb”.


DRIVERS

Surname	Name (Dim ..	
Hanks	Sam	Abc
Hawthorn	Mike	Abc
Herbert	Johnny	Abc
Hill	Damon	Abc
	Graham	Abc
	Phil	Abc
Hulme	Denny	Abc
Hunt	James	Abc
Ickx	Jacky	Abc
Ireland	Innes	Abc
Irvine	Eddie	Abc
Jabouille	Jean-Pierre	Abc
Jones	Alan	Abc
Kovalainen	Heikki	Abc
Kubica	Robert	Abc
Laffite	Jacques	Abc
Lauda	Niki	Abc
Leclerc	Charles	Abc
Maldonado	Pastor	Abc
Mansell	Nigel	Abc
Mass	Jochen	Abc
Massa	Felipe	Abc
McLaren	Bruce	Abc
Moss	Stirling	Abc
Musso	Luigi	Abc
Nannini	Alessandro	Abc
Nilsson	Gunnar	Abc
Ocon	Esteban	Abc
Pablo Montoya	Juan	Abc
Pace	Carlos	Abc
Panis	Olivier	Abc
Parsons	Johnnie	Abc
Patrese	Riccardo	Abc
Pérez	Sergio	Abc
Peterson	Ronnie	Abc
Piquet	Nelson	Abc

Figure 8: From "RESULTS.twb".