Formula 1 Data Analytics using a Graph-Based Approach

Author:
Ryan Brown

Supervisor:
Albert Burger

Second Reader:
Haslifah Hasim

A thesis submitted for the degree of
BSc (Hons) Computer Science

Heriot Watt University
Department of Mathematical and Computer Sciences

April 2021

Table of Contents

Abstract

Formula One is very much a team sport, even if the drivers get all of the recognition, as it requires thousands of hours of work by hundreds of different team personnel to even produce a car for the upcoming season. To get a car capable of winning the World Championship, millions of pounds must be spent to develop the car, and teams have an increasingly sophisticated number of devices designed to collect and analyse data to try to give them the advantage over the rest of the competition. Every factor which could potentially have an impact on the car, regardless of how minor it may seem, must be analysed thoroughly to ensure that nothing has been left behind. Graph databases are an increasingly popular method of visualising datasets, and the Neo4j tool is one of the most popular ways of working with graph databases, as it offers an intuitive insight into complex data. Created as a better alternative to the limitations of relational databases, graph databases allow the data to be stored using nodes, edges and properties to allow data analysis to be carried out. This project focuses on the graph database management system Neo4j, to look further at the use of graph databases and explore the use of graph-based data modelling, management and analysis using a Formula One based dataset.

Acknowledgements

My supervisor, Albert Burger, for all of his help and support in helping me to complete this project. Getting the opportunity to do a project like this for something I enjoy as much as Formula One has been amazing and it has been an amazing experience getting the chance to complete this project with his help.

My second reader, Haslifah Hasim, for their useful feedback on my first deliverable to allow me to complete my final project and this dissertation to the best of my ability.

Phil Bartie, for helping me to load in all of my F1 data by finding some alternatives to my local machine.

Tessa and Chris, for taking me to the British Grand Prix at Silverstone and giving me a wide variety of motorsport memorabilia to further increase my love of the sport of formula one.

My family, for their support throughout my time at university.

Declaration Statement

I, Ryan Brown confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of references employed is included.

Signed: Ryan Brown

Date: 16.4.2021

Introduction

If you want to be at the top of the sport, you need to have the most advanced technologies available to you, to gain a competitive edge over your rivals. Formula One teams use some of the most advanced data technologies in the world (MAPFRE, 2020) to try to gain some knowledge which can make them even slightly quicker than the other constructors. Without the ability to gather this huge amount of data in real-time, Formula One would be very different to the sport we know today. Every second a huge number of data points are generated in every area of the car, from the front wing to the engine, with countless engineers and designated data analysts working hard to turn this raw data into useful information which can be used to assist in crucial areas such as car development and race strategy. It is not just the 100 or so staff members who you see in the garages during a race weekend working hard to analyse this data, as each team will have a large number of highly talented staff working at their factory who are attempting to gain any slight advantage they possibly can. The ability to gather vast amounts of data, and put this into meaningful information, is absolutely crucial to the success of the time, as every point gained – or lost – can mean the difference of millions of pounds worth of prize money which, if things do not go as planned during a season, can lead to huge losses and possibly job losses and redundancies.

Aims and Objectives

The purpose of this research is to evaluate the use of graph-based technology to look at the benefits and limitations of the technology by exploring a Formula One dataset. This research will look at how graph-based modelling can be used for data analysis and if it is a viable choice which can be used when it is available. The overall aim of this project is to explore graph-based data modelling, management and analysis using a dataset related to Formula One. The project will take an F1 dataset (Ergast Developer API), convert it into a graph database within the database management system Neo4j and the data visualisation tool Neo4j Bloom, and use graph-based data modelling to analyse the

dataset, and the use of graph databases in data analysis. The main objectives that this project will look at are:

- Examine the benefits and limitations of the use of graph-based data modelling, and the Neo4j tool, to analyse a dataset
- Carry out queries to find useful information about the F1 dataset, and look at how the use of graph-based data modelling affects different types of queries being run
- Analyse the use of Cypher, the graph query language of Neo4j, and look at how easy (or difficult) it is to analyse queries with it. Discuss whether a certain type of problem or query is better suited to Cypher and how effective Cypher would be at analysing this dataset
- Explore the use of the Neo4j analysis tool, and the use of graph-based data modelling in general, to analyse datasets and how they can be used effectively in data analysis

Project Motivation

There is a huge amount of data being generated currently, and this amount increases hugely every single day (Batty et al, 2013). Whilst relational databases have been used for many years, they are often unable to cope when the amount of data inevitably increases, and the number of relationships grows (Jatana et al, 2012). These days, companies are beginning to investigate and use graph database technology instead, and it looks like this will be the future of big data which companies cannot afford to miss.

One of the most popular graph database tools is Neo4j, and there are many reasons why you should make the switch over from relational databases. As the amount of data increases, the number of relationships between the 'nodes' of data increases even more, and yet the performance of graph databases remains mostly constant even as the data increases. Traditional databases require you to use JOIN statements to compute relationships, which takes up huge amounts of time, whilst graph databases store the relationships alongside the data within the model. Being able to access all of this

data quickly is crucial to a company, and a graph database is designed to make this an extremely efficient process, theoretically allowing you to search through millions of relationships every second. Amongst many advantages which will be discussed throughout this project, graph databases are excellent at managing highly connected data and computing complex queries. One of the reasons for undertaking this project was to further explore how Neo4j, and the innovative graph database technology itself, can be used effectively to analyse a dataset, and what benefits the use of graph-based data modelling actually has.

Literature Review

The purpose of this project is to explore the use of graph databases and visualisation techniques to model a Formula One dataset in Neo4j, and explore the visualisations provided to analyse this dataset and discuss the advantages and disadvantages of choosing to use graph databases for the analysis of the dataset. This literature review will establish further understanding of the topics related to this project and will help to refine some of the future sections of my project.

History of Formula One

The Formula One World Championship first began in 1950, after years of unofficial Formula One cars racing at various circuits throughout Europe. These days the sport is much more of an international affair, with tracks being located in a wide range of countries from the United States to Abu Dhabi. Although plans for the official championship were discussed in the late 1930s, these did not go any further due to World War Two (Williamson, 2015). Discussions recommenced soon after the war and in May 1950 the first official race event was held at Silverstone, although unofficial events continued to run until 1983 when the increasing costs meant that it no longer made financial sense to run them.

Things started slowly, as due to a lack of participants the 1952 and 1953 seasons followed the rules of the lower F2 series, and many competitors left the sport as a result of the high costs involved. The technology of the cars continued to advance season after season, with cars initially be 4.5 litre front-engines, before new regulations were introduced which limited them to just 2.5 litres. Working as a team became even more important as the constructor's championship, won by the team with the most points, was introduced in 1958. Moving into the 1970s, cars were becoming even faster, as knowledge of aerodynamics improved to give huge amounts of downforce and therefore go even faster around high-speed corners. Later in the same decade turbochargers become popular, with the McLaren team benefitting hugely by winning 15 out of 16 races during the 1988 season, although restrictions were brought in (including banning the use of turbochargers) the following season. The world governing

body of the sport, the FIA, continued to make changes to try to make the racing more enjoyable to the spectator, as well as lower costs to encourage more teams to enter Formula One, and after the amazingly dominant spell of Michael Schumacher came to an end in 2006, the sport slowly became more competitive once again, and ensure it remains the pinnacle of motorsport to this day.

Although there were cases where drivers bought their own cars which they raced themselves, most of Formula One has been major manufacturers such as Ferrari, Alfa Romeo and Mercedes Benz. This shows just how important a team is to your overall success in the sport, as the minds of hundreds of team members is of course far better than having only a small number of people. Having designated sections of your team analysing the huge range of data, with the amount of data increasing hugely each year, gave you a huge boost to your championship hopes as you would often be able to predict your race strategy, but also crucially what your opponents might do.

Early Days of Data in F1

In the early days of Formula One, the success of a race for a team was decided solely by the driver and the decisions they made out on the track (Wooden). The team simply manufactured the car, gave it to the driver, and that was as far as their contribution went. They could not look at any data such as tyre wear or engine modes whilst out on track like nowadays, and it was down to the driver to give feedback after the race weekend so that the team can make any changes which we required for the next event. If the driver did not feel like he could give proper feedback to the rest of the team, then that was that, and they would use a car which was pretty much identical next weekend. Even something as simple as lap times were far less accurate than now, as they were simply measured using stopwatches, with each car being designated one stopwatch and someone in race control having to physically stop the watch when the driver had finished their lap.

It wasn't until the 1990s that Formula One had progressed towards basic computers which could gather dozens of data points during the testing sessions, which was a huge improvement but did not come close to what is being achieved today.

Data Analysis in F1 Today

These days in Formula One does not even compare to those of just a few decades ago, as an unbelievably large amount of data is generated by every single car, and every driver, during the race weekend of a Grand Prix (MAPFRE, 2020). Even though the previous cars, like every other thing in the world, generated large amounts of data, it was not until very recently that teams had the tools and knowledge to be able to analyse this data and learn valuable information about it that we see today.

An individual Formula One car can often have up to 300 sensors on it, with each one generating millions of unique points of data at any time they are out on track. The cars themselves can end up with approximately 300GB of data each, which when combined with the rest of the team can end up being 50TB (terabytes) of data which the engineers and data analysts must go through to be able to seek any advantage they possibly can over their rivals. Being able to handle this data is an exceptional feat by itself, with teams requiring the most advanced technology and huge amounts of computing power to be able to continue their data analysis in real-time.

InfoZoom

InfoZoom is a data analysis and visualisation tool which was created by humanIT in 1997 and allows users to extract large amounts of data from a variety of different sources. InfoZoom displays database relations in tables with the rows being attributes and the objects as columns, and it allows users to explore the different data visualisations to try to gain a better understanding of the data being analysed.

This tool can be used to model certain Formula One datasets (Spenke and Beilken), and in one such example a column represents the participation of a driver in a certain race. The database being analysed contains data of races between the 1978 and 1998 seasons, with some drivers having their picture included in the dataset. InfoZoom allows for a compressed version of the dataset, as the dataset example contains approximately 8000 columns, accommodating attributes such as driver name, team, race location and finish position. InfoZoom is able to compress this table, which can be considered overwhelming, by reducing the width of the column until every column can fit on the screen at the same time, which means that each column now has a width of about 0.1 pixels. In order to make this highly compressed data readable, special techniques are used by the software, such as neighbouring cells which have identical values being combined into one larger cell. Although the use of InfoZoom could be explored further, this report will not focus on this tool as the scope of the project would be too large, and instead this project will focus on the data visualisation.

Graph Databases

Graph databases are databases which use a graph structure in order to represent and store data with nodes and edges and allow queries to be run on this data. The edges of the graph database represent the relationships between the nodes, and these relationships can be visualised easily which makes graph databases very useful for heavily connected data. Graph databases were introduced to try to get around the limits of relational databases, as relationships can be labelled, directed and given various properties, and graph databases are designed to treat these relationships as being just as important as the data itself (Neo4j, 2019). Relational databases often have to compute the relationships at runtime, which can take a long time, whilst graph databases store the data and relationships together in the model which makes accessing nodes and relationships much more efficient and allows you to search through millions of relationships every second.

Nodes are the entities in a graph database, and they can hold any number of attributes you would like which are called properties, as well as being able to create labels for them. Relationships provide directed, named connections between these nodes of data, such as Driver RACED_FOR Team, and all relationships must connect 2 nodes, and have a certain direction between them. One of the main benefits of using graph databases is the increased performance (Neo4j, 2018), as your dataset continues to get larger, the number of relationships will increase exponentially. In a graph database the performance level will stay the same, whereas a relational database will continue to slow down substantially as the number of relationships increase. There is no limit to the number of relationships a node can have (AWS, 2017), and graph databases are able to be so efficient as the relationships are stored together with the nodes of data in the model rather than being calculated at runtime. They are often used in social networking or recommendation engines, as it allows the company to make recommendations of what products a customer should buy, simply by looking at the purchase history of other customers who have similar interests to themselves.

Neo4j

Neo4j is an open-source graph database with source code that was written in Java and Scala and is currently being used by thousands of different companies in a range of different areas, including government, manufacturing and financial sectors. Neo4j uses the declarative language of Cypher (Holzschuher and Peinl, 2013) to write queries and access the data, which can be considered quite similar to Structured Query Language (SQL), which often makes it easy for developers to begin using Neo4j as their tool of choice. Cypher also allows you to manipulate the data using UPDATE and DELETE statements, similar to SQL. However, Neo4j should be avoided if the dataset does not have relationships involved (Nayak et al, 2013). The Neo4j software is split into three categories (Guia et al, 2017), with the Community edition being available for everyone to use, and the Enterprise edition more suited to commercial purposes which includes having online backups, a detailed monitoring system and a strong lock management system for the database. There is also a Government edition available, which contains even more features than the Enterprise edition, which is more focused on

government services. The interface is simple and easy to use, which makes it accessible to almost anyone who may currently be using relational databases and wants to consider upgrading their software to increase efficiency during query execution.

Graph databases are an efficient method of storing large amounts of information which are strongly interconnected (Fernandes and Bernardino, 2018), and graph databases are well optimised for storing and querying data in graphs. However, although they are an efficient method of querying large amounts of strongly connected data, they are much less efficient at some other operations (Vicknair et al, 2010). Neo4j is considered a popular choice for graph databases as it uses Cypher, which is a powerful query language, and also has an intuitive interface which allows users to access the required information quickly. As graph databases have a dynamic schema which does not need to be predefined, they are much more flexible than relational databases (Sahatqija et al, 2018) which means that they are more suitable to environments which are constantly evolving. However, security is also a very important issue for databases. Traditional relational databases have some very secure methods in place to ensure good security, and there is a concern over the privacy and security of the data that is stored in the graph database.

Although traditional relational databases such as MySQL have been used for a long time, graph databases, with Neo4j being a common choice, have been becoming an increasingly popular method of representing data. Graph databases are known as NoSQL, as they are schemaless databases, and do not usually have a declarative query language (Have and Jensen, 2013). Neo4j is an exception to this, as it uses the Cypher query language, and is continuing to develop and evolve. Taking a simple example of a neighbour network, finding the immediate neighbours of a node and the interactions between them, the use of Cypher and graph databases can be about 36 times faster than running a simple SQL query, although it could be quicker using PostgreSQL by simply decomposing a complex query into a number of simple SQL queries. Also, for finding the shortest path between two nodes, Cypher actually has a dedicated function for this which can mean that it is much quicker at computing the shortest path when compared to a traditional relational database such as PostgreSQL. It is however

important to consider that a graph database is not necessarily always the best choice, and it often depends on the type of query you need. Graph databases are often quicker for queries using paths (e.g., the shortest path between two nodes) instead of needing to use expensive JOIN commands, and relational databases will often be better with set operations, as graph databases have not yet been optimised for these sorts of queries.

There is a data visualisation tool within Neo4j which is called Neo4j Bloom (Neo4j), which can be accessed via the Neo4j desktop software. It is available for free for local databases instances with no activation or download being required. Neo4j Bloom is a data exploration tool which allows users to visually explore their graph databases from a variety of different perspectives, and allows the work completed to be shown to many different people involved in the project, regardless of their expertise. Neo4j Bloom makes it possible to write patterns to retrieve data and traverse complicated graphs. For example, if you had a graph database which contained orders, customers, products and suppliers, it would be possible to easily find suppliers who supply more than one product. This can be done using a palindrome search, by typing 'Product Supplier Product' which would then return the desired results, without needing to specify the relationships (although you can choose the relationships from the suggestions made by Neo4j Bloom).

Currently, there is no absolute answer to whether you should choose a traditional relational database, or a graph database, to store your data. Graph database technologies are often considered a better alternative to relational databases for certain requirements (Vicknair et al, 2010), such as when processing specific query types to find out a particular insight, although there are definitely still some areas where graph databases do not provide the same levels of efficiency and performance that traditional methods are able to provide. It is important that you consider a variety of different factors before making your decision, such as the flexibility, performance and security which is required. Graph databases are being developed and improved at an increasing rate, and some of the disadvantages currently being discussed may soon be changed which can make the decision to stay with traditional methods even harder to justify.

The F1 Data Set

This section describes the dataset which will be used throughout the project. The data that will be analysed is the Ergast Developer API[1], which is a comprehensive web service that provides a wide range of Formula One Grand Prix data starting from the first ever World Championship in 1950.

This dataset is free to use and is available on the internet for anyone to utilise. From the terms and conditions page[2], it clearly states that the author gives you permission to use this web service for any personal, non-commercial applications and services, which includes research and educational purposes. The only condition is that you must not charge for any application or service which makes use of the web service, although use on websites supported by advertising is permitted.

Retrieving the Data

This API service allows for you to get the data via a GET request, using the following syntax:

```
http[s]://ergast.com/api/<series>/<season>/<round>/...
```

where:

- `<series>` - should be set to ‘f1’
- `<season>` - a four-digit integer
- `<round>` - a one- or two-digit integer

The API can display the results of the query in either XML, JSON or JSONP formats. The default is XML, or when ‘.xml’ is appended to the end of the URL. JSON is obtained by appending ‘.json’ to the URL, and JSONP is obtained by appending ‘.json’ and a query parameter named ‘callback’ which needs to specify a valid JavaScript function name. Another option to use the dataset is to download the tables in CSV format to allow for it to be imported into data analysis software, such as Microsoft Excel and Neo4j.

[1] https://ergast.com/mrd/
[2] https://ergast.com/mrd/terms

Description of the Data Set

This section will cover all of the different tables and fields within the dataset and give descriptions about what each part of the dataset describes. Full descriptions about the dataset can be found within the appendix (Appendix A), and a diagrammatic form of the data set can also be found in the appendix (Appendix C). This diagram allows you to see how all of the data is linked together and provides an overview of the dataset which gives a better understanding of what it contains.

The first column of all tables in the dataset is a unique identifier for each specific row (record) in the table.

circuits.csv

This table covers all of the circuits which have been raced at throughout the history of Formula One, since 1950. It contains information on the name of the circuit, the city and country it is located in, the precise latitude and longitude, and URL links to find some more information.

Data Examples:

1, albert_park, Albert Park Grand Prix Circuit, Melbourne, Australia, -36.8497, 144.968, 10, http://en.wikipedia.org/wiki/Melbourne_Grand_Prix_Circuit

14, monza. Autodromo. Nazionale Di Monza, Monza, Italy, 45.6156, 9.28111, \N, http://en.wikipedia.org/wiki/Autodromo_Nazionale_Monza

constructors.csv

This part of the dataset covers all of the constructors (teams) who have taken part in Formula One at some point in history. It contains the name of the constructor, the nationality of where they originated, and URL links containing more information.

Data Examples:

9, red_bull, Red Bull, Austrian, http://en.wikipedia.org/wiki/Red_Bull_Racing

23, brawn, Brawn, British, http://en.wikipedia.org/wiki/Brawn_GP

105, maserati, Maserati, Italian, http://en.wikipedia.org/wiki/Maserati

constructor_results.csv

This piece of data links the race which took place and the constructor who took part and displays the number of points gained by the constructor, and the status of them in the race.

Data Examples:

Unique identifier, race identification, constructor identification, points, status

11, 18, 11, 0, \N

413, 56, 13, 0, \N

2182, 222, 24, 0, \N

constructor_standings.csv

This part of the dataset collects all of the positions, and number of points, of all of the different constructors who have taken part in a race. It contains the unique identifiers of the race which took place and the constructor who took part, to link them together, as well as the number of points they gained and what championship position they are currently in. The final column indicates the number of wins.

Data Examples:

unique identifier, race identification, constructor identification, points, position, position in text format, wins

3, 18, 3, 9, 2, 2, 0

45, 22, 6, 63, 1, 1, 4

25783, 40, 11, 0, 9, 9, 0

drivers.csv

This data contains information on all of the drivers who have taken part in Formula One, including their full name, unique driver number (only drivers from the 2014 season onwards have chosen their own number), date of birth, nationality and a URL to find further information.

Data Examples:

17, webber, \N, WEB, Mark, Webber, 27/08/1976, Australian, http://en.wikipedia.org/wiki/Mark_Webber

154, grosjean, 8, GRO, Romain, Grosjean, 17/04/1986, French, http://en.wikipedia.org/wiki/Romain_Grosjean

races.csv

This section of the dataset categorises all of the races which have taken place (unlike circuits.csv which just shows the actual tracks). It shows the year of the race, which circuit the race took place on, the date and time it took place, and a URL where more information can be found about the race events.

Data Examples:

Unique identification, year, round number, circuit identification, name, date, time, URL

20, 2008, 3, 3, Bahrain Grand Prix, 06/04/2008, 11:30:00,

http://en.wikipedia.org/wiki/2008_Bahrain_Grand_Prix

430, 1985, 11, 39, Dutch Grand Prix, 25/08/1985, \N,

http://en.wikipedia.org/wiki/1985_Dutch_Grand_Prix

lap_times.csv

This comprehensive part of the dataset displays every lap completed by each driver at every race they took part in. It links the race and driver together using their unique identifiers, and shows the lap number, current position they were in, and the lap time.

Data Examples:

Race identification, driver identification, lap number, driver position, lap time, lap time in milliseconds

842, 13, 16, 8, 01:44:1, 104098

964, 817, 1, 5, 01:41:7, 101724

pit_stops.csv

This section of the dataset shows all of the pitstops which have taken place throughout all of the races. It uses the unique identifiers of the driver and race to link them together, the stop number of the driver (e.g. their first stop of the race), the lap number of the pitstop, the time that the stop took place and how long the driver was in the pitlane for.

Data Examples:

Race identification, driver identification, stop number, lap number, time of stop, length of stop, length of stop in milliseconds

848, 15, 2, 34, 15:05:36, 22.131, 22131

991, 817, 2, 31, 15:05:15, 22.691, 22691

qualifying.csv

This piece of the dataset displays all of the qualification results of all race events. It uses the unique identifiers of the race, driver and constructor to link them together correctly, and shows the number, qualifying position and the final times from all 3 qualifying sessions.

Data Examples:

Unique identification, race identification, driver identification, constructor identification, number, race starting position, 1st session time, 2nd session time, 3rd session time

19, 13, 6, 2, 1, 01:35.3, 01:34.4, 01:35.7

5715, 893, 821, 15, 12, 10, 01:45.5, 01:44.2, \N

seasons.csv

This dataset simply classifies each season which has taken place. It contains the year and a URL for further information about the season.

Data Examples:

1978, https://en.wikipedia.org/wiki/1978_Formula_One_season

1955, https://en.wikipedia.org/wiki/1955_Formula_One_season

2015, https://en.wikipedia.org/wiki/2015_Formula_One_season

status.csv

This data collates all of the different finishing statuses of the drivers, for example if they have been disqualified or had a brakes failure which caused them to not finish the race.

Data Examples:

3, Accident

27, Tyre

60, Out of fuel

driver_standings.csv

This section of data gathers all of the points gained by drivers at each race event. It links the race and driver by their unique identifiers, and shows the number of points they have, their position, and number of wins.

Data Examples:

Unique identification, race identification, driver identification, number of points, position, position in text format, number of wins

23, 19, 21, 0, 15, 15, 0

245, 30, 1, 76, 1, 1, 4

results.csv

This dataset collects all of the data from a race event and collates it into this 1 set. It contains unique identifiers for the race, driver and constructor to link them together, and shows the number, grid position of the driver, the number of points gained, the number of laps the driver completed, the total race time taken, when they completed their fastest lap of the race, their fastest lap time and speed, and their finishing status.

Data Examples:

Unique identification, race identification, driver identification, constructor identification, number, grid position, finish position, finish position in text format, finish position without specifying if they finished or not, the number of points gained, number of laps completed, total time taken to finish, total time in milliseconds, what lap was their fastest, ranking based on fastest laps, their fastest lap time, maximum speed during their fastest lap, and their finishing status,

1, 18, 1, 1, 22, 1, 1, 1, 1, 10, 58, 34:50.6, 5690616, 39, 2, 01:27.5, 218.3, 1

66, 20, 20, 5, 15, 19, \N, R, 22, 0, 0, \N, \N, \N, \N, \N, \N, 5

24625, 1030, 840, 211, 18, 12, \N, R, 20, 0, 45, \N, \N, 25, 17, 01:43.3, 193.507, 23

Research Methodology

In this section the requirements of the project are set out, to show what should be achieved when completing this project, and to prioritise what needs to be done. The project aims to explore the use of graph databases further to explore a large dataset, to try to gain a further understanding of the graph database tool Neo4j, as well as the advantages and disadvantages of using graph-based data modelling to analyse these datasets.

Project Requirements

The overall aim of this project is to discuss the use of graph-based data modelling to analyse a dataset by using the Neo4j tool. The first requirement of this project would be to insert the entire dataset into Neo4j, so that it is ready to be analysed. This would involve ensuring that all of the data values, and relationships, are added correctly to ensure that the data has been setup and analysis can be carried out successfully. Any challenges which were faced during this will be discussed throughout the project, as well as any parts which were made easier due to the use of Neo4j.

Another requirement of this project is to explore the use of graph-based data modelling when analysing large datasets and discuss the suitability of using it for this data analysis. This will involve carrying out queries (a sample of which can be found in section 3) to look at how effective Neo4j is at analysing the dataset, but also to find some useful information about Formula One at the same time. Any challenges which may arise when carrying out this objective will be discussed throughout the project, as well as discussing any benefits to graph-based data modelling. By carrying out a wide range of different queries which have been categorised appropriately, it should be possible to evaluate both the advantages and disadvantages to using Neo4j for certain types of query, as it may be better suited to one type of query compared to another type.

The project will also have a requirement to analyse the data analysis tool of Neo4j, to explore further how it can help to analyse datasets and how they can be used effectively for data analysis. This

project will look at which query types are most suitable to being run in Neo4j, and for which ones Neo4j brings no real benefit to. This requirement will involve examining how Neo4j performs when:

- Loading in all of the data – can it handle large volumes of data, including all of the nodes and relationships which must be included and displayed in the graph format? Also, were there any challenges, or benefits, faced when loading in this data?
- Analysing queries – how effective is Neo4j at running a variety of different query types? Is it more suited to a certain type of query, or it is able to effectively handle many different types?
- Data visualisation – by using the data visualisation tool Neo4j Bloom, the project will visually explore the F1 dataset, retrieving data and traversing the graph database, to try to analyse which types of queries are best suited to Neo4j Bloom. The project will look at which query types would be benefitted by the use of Neo4j Bloom.

Requirements Completed

This project was largely successful in its requirements of analysing a Formula One dataset to look at the use of graph-based data modelling and the Neo4j tool at analysing large datasets. When it came to the loading in of the data, this is clearly not one of the strong features of Neo4j as it took a very long time for the data to be loaded into the software. For the lap times file, which contains the details of every lap raced in F1, and is over half a million rows, it took approximately 3 days to be completely loaded in. However, when it came to the queries and the data visualisation, this is where Neo4j really proved how useful it can be as it showed many advantages over relational databases.

Data Creation and Querying in Neo4j

In order for this project to be completed in its entirety, the first priority was to get all of the data (in the form of csv files) loaded into to the Neo4j graph database, something which would not be an easy task as some of the files were relatively large. As well as all of the data, relationships needed to be added which would also take a long time to fully be completed. After all of the data was loaded in, the queries needed to be run to analyse the dataset and the Neo4j software, before then looking at the data visualisation and how it appears within Neo4j. This section details all of the work that has been carried out for this project.

Data Insertion

When it came to inserting the data from the csv files into the Neo4j graph database, an issue arose as a result of the very large quantity of data being inserted. The computer, a 2018 MacBook Air, was unable to cope with the vast amount of data which was being attempted to be inserted with the local Neo4j instance and was therefore only able to insert about half of the data with this method.

The research carried out as a result of this led to a number of possible solutions which could be used in order to ensure that all of the data was loaded in successfully.

1. Run it on the virtual machine supplied by the university – a more powerful computer which would probably be capable of carrying out the required data insertions. However, the download and installation of this required quite a lot of available storage and RAM, which was simply not available, so this option was not suitable
2. Run it on the university computers locally – this is an ideal solution as it would give all of the benefits of the virtual machine (option 1) without requiring any download or installation, but due to the COVID-19 pandemic strict travel restrictions and stay-at-home orders were put in

place by the government which meant that it was not possible to travel into university to use this option

3. Use the X2GO service to be able to use a (more powerful) university computer on a server – this is similar to the virtual machine but did not require as high memory installation requirements. However, this is quite slow to use and is not an ideal solution
4. The final option was to use the Google Collab tool (tutorialspoint) – this is a free environment which runs entirely in the cloud and does not require any setup. Although designed primarily for use with the Python programming language, it can also be used with Cypher, the Neo4j language. To save and edit documents, only a Google account is required and can be edited by multiple users, similar to the word processing software Google Docs. This was the ideal choice for this project as it allowed for the use of much more powerful computers with almost zero setup, and can be resumed quickly and easily

When it came to actually loading in the data, the majority of csv files were inserted into Neo4j without too many problems, and the code to do this can be found in appendix F. Although my local machine was unable to handle the large data files, the Google Collab software worked effectively, and all-but-one of the files were inserted successfully. However, the lap_time.csv file (which contains the details of every single lap ever recorded in Formula One) proved to be troublesome as it contained almost half a million (500,000,000) lines which proved to be extremely large and time-consuming to load data into Neo4j. Table 1 shows the time taken for the lap_times.csv file to be loaded into an empty Neo4j database (no other files had been inserted) using the Google Collab software which has 12GB of RAM available to use:

Number of Entries	Time Taken to Load Data
1000	4.6 seconds
5000	20.3 seconds

10,000	64.3 seconds
25,000	356.8 seconds (~6 minutes)
50,000	1228.9 seconds (~20 minutes)
100,000	4600.9 seconds (~76 minutes)
250,000	~8 hours
Entire File (~500 ,000)	Unknown

Table 1: Time Taken to Load in lap_time.csv (Google Collab)

As can be seen from the table, the time taken to load in the data does not increase in a linear fashion as the number of entries increases. Instead, it increases in an exponential manner which it makes it very difficult and awkward to be able to insert the data fully. In fact, it was not actually possible to insert the entire data file into the database using Google Collab as based on the previous results from the table, it would have taken far too long for the data insertion to be completed. Despite all of this, it is possible to monitor the RAM utilisation throughout the cloud session of Google Collab and it never got close to the 12GB that had been allocated to the session, which suggests that the amount of memory being used was not really ever an issue.

The other option was to use x2go to run the data loading commands overnight, using the screen command on Linux, to allow the file plenty of time to be inserted into the Neo4j database. After some slight troubleshooting issues to work out how this would be possible, the issues were eventually fixed, and the process was as follows:

- Use the `screen` command to allow for the Neo4j commands to run in the background, even when not logged on to the server
- SSH into the Heriot Watt student server, where Neo4j is installed
- Load in the data files, using cypher-shell and LOAD CSV

- Exit the screen and logout, waiting overnight for the data to be loaded into the database
- Log back into the server, using `screen -r` to enter the previous session

Therefore, the final solution that was chosen in order to ensure all of the nodes and relationships were loaded in fully and successfully was the Heriot Watt university server. It was agreed that, to ensure the server did not end up slowing down too much to the detriment of others who may need to use it, that the server that would be used would be one that was not currently in use, which meant that the Cypher commands could be run overnight without needing to worry how it would affect other users. The cypher commands for loading in the dataset, including all nodes and relationships, is:

```
// load in the drivers.csv file, changing the variable names – 848
nodes
LOAD CSV WITH HEADERS from 'file:///drivers.csv' AS row MERGE
(d:Driver {driverId: row.number, forename: row.forename, surname:
row.surname, code: row.code, nationality: row.nationality,
driverRef: row.driverRef, driverId: row.driverId, dob: row.dob, url:
row.url}) RETURN d;

// load in the constructors.csv file, changing the variable names –
210 nodes
LOAD CSV WITH HEADERS from 'file:///constructors.csv'
AS row WITH row WHERE row IS NOT NULL
MERGE (c:Constructor {constructorId: row.constructorId,
constructorRef: row.constructorRef, constructor: row.name,
nationality: row.nationality, url: row.url})
RETURN c;

// load in the seasons.csv, changing the variable names – 71 nodes
LOAD CSV WITH HEADERS from 'file:///seasons.csv' AS row MERGE
(s:Season {year: row.year, url: row.url}) RETURN s;

// load in the status.csv, changing the variable names – 135 nodes
LOAD CSV WITH HEADERS from 'file:///status.csv' AS row MERGE (f:
finishingStatus {statusId: row.statusId, status: row. Status})
RETURN f;

// load in the circuits.csv, changing the variable names – 74 nodes
LOAD CSV WITH HEADERS from 'file:///circuits.csv'
AS row
MERGE (t: track {circuitId: row.circuitId, circuitRef: row.
circuitRef, name: row.name, location: row.location, country:
row.country, latitude: row.lat, longitude: row.lng, alt: row.alt,
url: row.url})
RETURN t;

// load in the races.csv, changing the variable names – 1031 nodes
```

```
LOAD CSV WITH HEADERS from 'file:///races.csv'
AS row
MERGE (r: race {raceId: row.raceId, year: row.year, round:
row.round, circuitId: row.circuitId, name: row.name, date: row.date,
time: row.time, url: row.url})
RETURN r;

// add a relationship between circuits.csv and races.csv
// note: use MERGE instead of CREATE to avoid creating the same
relationship twice if run multiple times
MATCH (t:track),(r:race)
WHERE t.circuitId = r.circuitId
MERGE (t)-[rel:RACED_AT]->I
RETURN type(rel)

// load in the constructor_results.csv, changing the variable names
- 11,560 nodes
LOAD CSV WITH HEADERS from 'file:///constructor_results.csv'
AS row
MERGE (cr: constructorResults {constructorResultsId:
row.constructorResultsId, raceId: row.raceId, constructorId:
row.constructorId, points: row.points, status: row.status})
RETURN cr;

// add a relationship between constructor results, constructors and
races - 23120 relationships
MATCH (cr:constructorResults),(c:Constructor),(r:race)
WHERE cr.raceId = r.raceId AND cr.constructorId = c.constructorId
MERGE I-[rel2:FINISH_POSITION]->(cr)
MERGE I-[rel3:TEAM_POINTS]->(cr)
RETURN type(rel2), type(rel3)

// load in the constructor_standings.csv, changing the variable
names - 12326 nodes
LOAD CSV WITH HEADERS from 'file:///constructor_standings.csv'
AS row
MERGE (cs: constructorStandings {constructorStandingsId:
row.constructorStandingsId, raceId: row.raceId, constructorId:
row.constructorId, points: row.points, position: row.position,
positionText: row.positionText, wins: row.wins})
RETURN cs;

// add a relationship between constructor standings, constructors
and races - 24632 relationships
MATCH (cs:constructorStandings),(c:Constructor),(r:race)
WHERE cs.raceId = r.raceId AND cs.constructorId = c.constructorId
MERGE I-[rel4:TEAM_POSITION]->(cs)
MERGE I-[rel5:TEAM_FINISH]->(cs)
RETURN type(rel4), type(rel5)

// load in the pit_stops.csv, changing the variable names - 7436
nodes
LOAD CSV WITH HEADERS from 'file:///pit_stops.csv'
AS row
```

```
MERGE (pt: pitStops {raceId: row.raceId, driverId: row.driverId,
stop: row.stop, lap: row.lap, time: row.time, duration:
row.duration, milliseconds: row.milliseconds})
RETURN pt;

// add a relationship between pitstops, drivers and races - 14872
relationships
MATCH (pt:pitStops),(d:Driver),(r:race)
WHERE pt.raceId = r.raceId AND pt.driverId = d.driverId
MERGE (d)-[rel10:PITTED]->(pt)
MERGE I-[rel11:STOPPED]->(pt)
RETURN type(rel10), type(rel11)

// load in the qualifying.csv, changing the variable names - 8354
nodes
LOAD CSV WITH HEADERS from 'file:///qualifying.csv'
AS row
MERGE (qt: qualifying {qualifyId: row.qualifyId, raceId: row.raceId,
driverId: row.driverId, constructorId: row.constructorId, number:
row.number, position: row.position, q1: row.q1, q2: row.q2, q3:
row.q3})
RETURN qt;

// add a relationship between qualifying, drivers, races and
constructors - 25062 relationships
MATCH (qt:qualifying),(d:Driver),(r:race), (c:Constructor)
WHERE qt.raceId = r.raceId AND qt.driverId = d.driverId AND
qt.constructorId = c.constructorId
MERGE (d)-[rel12:START_POSITION]->(qt)
MERGE I-[rel13:QUALI_TIME]->(qt)
MERGE I-[rel14:TEAM_STARTING]->(qt)
RETURN type(rel12), type(rel13),  type(rel14)

// load in the results.csv, changing the variable names - 24620
nodes
LOAD CSV WITH HEADERS from 'file:///results.csv'
AS row
MERGE (rr: results {resultsId: row.resultsId, raceId: row.raceId,
driverId: row.driverId, constructorId: row.constructorId, number:
row.number, grid: row.grid, position: row.position, positionText:
row.positionText, positionOrder: row.positionOrder, points:
row.points, laps: row.laps, time: row.time, milliseconds:
row.milliseconds, fastestLap: row.fastestLap, rank: row.rank,
fastestLapTime: row.fastestLapTime, fastestLapSpeed:
row.fastestLapSpeed, statusId: row.statusId})
RETURN rr;

// add a relationship between results, drivers, races, constructors
and status - 98480 relationships
MATCH (rr:results),(d:Driver),(r:race), (c:Constructor),
(f:finishingStatus)
WHERE rr.raceId = r.raceId AND rr.driverId = d.driverId AND
rr.constructorId = c.constructorId AND rr.statusId = f.statusId
MERGE (d)-[rel15:DRIVER_FINISH]->(rr)
MERGE I-[rel16:RACE_FINISH]->(rr)
MERGE I-[rel17:TEAM_FINISH]->(rr)
```

```
MERGE (f)-[rel18:ENDED_RACE]->(rr)
RETURN type(rel15), type(rel16),  type(rel17), type(rel18)

// load in the driver_standings.csv, changing the variable names -
32586 nodes
LOAD CSV WITH HEADERS from 'file:///driver_standings.csv' AS row
MERGE (ds: driverStandings {driverStandingsId:
row.driverStandingsId, raceId: row.raceId, driverId: row.driverId,
points: row.points, position: row.position, positionText:
row.positionText, wins: row.wins}) RETURN ds


// add a relationship between driver standings, drivers and races -
65132 relationships
MATCH (ds:driverStandings),(d:Driver),(r:race) WHERE ds.raceId =
r.raceId AND ds.driverId = d.driverId MERGE (d)-
[rel6:CHAMPIONSHIP_POSITION]->(ds) MERGE I-
[rel7:FINISH_DRIVER_POSITION]->(ds) RETURN type(rel6), type(rel7)

// load in the lap_times.csv, changing the variable names - about
half million nodes (takes 3-4 days)
USING. PERIODIC COMMIT 25000
LOAD CSV WITH HEADERS from 'file:///lap_times.csv'
AS row
MERGE (lt: lapTimes {raceId: row.raceId, driverId: row.driverId,
lap: row.lap, position: row.position, time: row.time, milliseconds:
row.milliseconds})
RETURN lt;

// add a relationship between laptimes, drivers and races
MATCH (lt:lapTimes),(d:Driver),(r:race)
WHERE lt.raceId = r.raceId AND lt.driverId = d.driverId
MERGE (d)-[rel8:LAPPING_AT]->(lt)
MERGE I-[rel9:TIMED_LAP]->(lt)
RETURN type(rel8), type(rel9)
```

By using the university server to load in the dataset, the commands could be set to run overnight until they were completed without needing to actually watch them being inserted into the database. This meant that almost all of the commands were loaded successfully, including the file which contained every single lap time in Formula One by every driver during a race, which ended up taking approximately 3-4 days to be loaded in fully.

The only issue was when trying to insert the relationships between the lap times, drivers and races, as even when the command was left to run in the background, it had still failed to be successfully loaded after 2 full weeks of being run. When loading in a csv file, there are a number of things which can be changed to load in some data quicker, such as by using periodic commit which instructs Neo4j to

perform a commit after a specified number of rows, which therefore reduces the memory overhead of the transaction state. The default is for a commit to occur every 1000 rows, although for a large file such as lap_times.csv it was changed to force a commit every 25000 rows. Another option for loading in large amounts of data is the neo4j-admin import tool, although it is important to note that it can only be done once in an unused database. Although this does require some work to be done on the csv files beforehand, it can quite substantially reduce the amount of time taken to load in the data file. However, whilst these solutions will help when loading in a csv file to add nodes to a database quite effectively, adding the relationships of a graph database between nodes does not involve the importing of a csv file. Therefore, it is not possible to use the periodic commit or neo4j-admin import tools to help reduce the time and so there must be some other method which could be used to reduce the amount of time to load in the relationship. Interestingly, the lap_times.csv file is approximately half a million rows in size, which in the context of graph databases and Neo4j is not actually particularly large, as some graph databases can contain multiple millions of nodes and even more relationships between these nodes. Therefore, why does this specific dataset seem to take such a long time to fully be loaded in? Is the dataset organised in a particularly inefficient way? Further research will be required into this area to work out why this formula 1 dataset is taking so long to load into the Neo4j graph database, as it seems peculiar why this relatively modest dataset is taking much longer than another unrelated dataset which contains millions more nodes and relationships.

Another potential solution was eventually found after undergoing some research on the topic to be able to load in the lap times relationships. The nodes and relationships would be loaded in at the same time in the same command, which would allow for PERIODIC COMMIT to be used to reduce the amount of memory required. Although CREATE would have been quicker than using MERGE, it was important that there were no duplicates within the dataset, so this extra memory usage was considered important. As well as this, the creation of indexes was introduced on the races and drivers' nodes to try to speed up the insertion of the relationships. Although using indexes does require more memory initially, and it takes time to create them, it can substantially improve node lookup performance and make queries quicker which can often make them a good choice to introduced into your graph

database. However, the decision on which indexes to use is often non-trivial as it does use up extra memory when they are created. For queries will be used frequently, indexes will bring a large benefit to the speed of the data insertion and query processing, although you should still carefully consider if an index is actually required.

```
CREATE INDEX driverIndex FOR (d:Driver) ON (d.driverId);
CREATE INDEX lapIndex FOR (r:race) ON (r.raceId);

CREATE INDEX lapCompositeIndex
FOR (lt:lapTimes)
ON (lt.raceId, lt.driverId, lt.lap)


// load in the lap_times.csv, changing the variable names
// 472941 nodes and 945008 relationships
USING PERIODIC COMMIT 25000
LOAD CSV WITH HEADERS from 'file:///lap_times.csv' AS row
MERGE (d:Driver {driverId:row.driverId})
MERGE (r:race {raceId:row.raceId})
MERGE (lt:lapTimes {raceId: row.raceId, driverId: row.driverId, lap:
row.lap, position: row.position, time: row.time, milliseconds:
row.milliseconds})
MERGE (d)-[:LAPPING_AT]->(lt)
MERGE I-[:TIMED_LAP]->(lt)
```

The range of queries which have been created have been designed to test how effective Neo4j is at analysing a dataset using queries and the Cypher language. For each of the queries, the structure of the query is discussed as well as the output of the query. The queries which are being analysed can be found in Appendix D. All of the queries did run successfully, and the results of the three iterations are discussed in the next sections.

Test Queries

In order to test that the data had been loaded correctly, some queries (other than the sample queries found in appendix D) were run to check the output and see if anything unexpected would be returned. Queries were run to find out some interesting data which is detailed in this chapter.

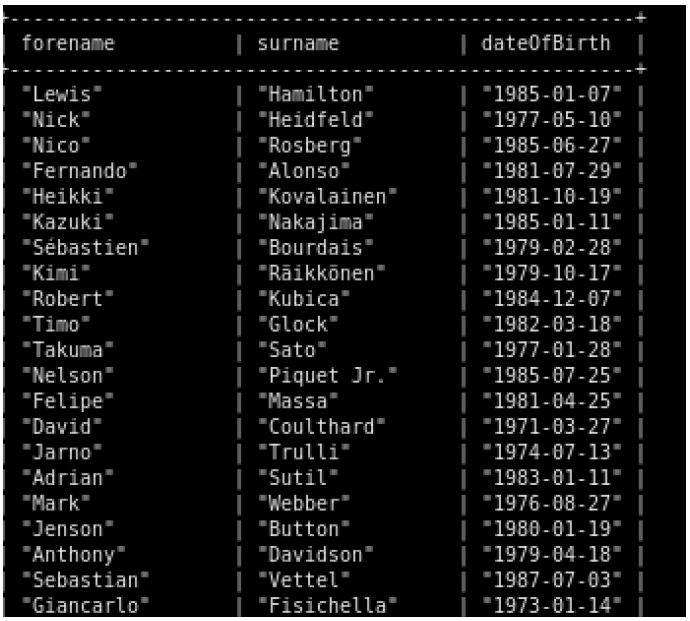
Test Query 1 – Return All Drivers

This query returns the full names and date of births of all drivers who have ever raced in Formula One. The Cypher language, used in Neo4j, is similar to Structured Query Language (SQL) as it uses much of the same structure, which could make it a relatively easy language to learn if you decided to switch from a relational database system to graph database modelling software such as Neo4j. This is one of the simplest queries to construct as it only looks at the set of drivers in the dataset. The MATCH keyword starts every query in Cypher, and it is used to specify which parts of the data will be used during this query (i.e., the driver nodes). As this query asks for every driver, the output does not need to be filtered so the RETURN statement contains the full names and date of birth fields of the drivers. The AS keyword simply allows you to rename the column headings to a name of your choice, so this query statement has the column readings forename, surname and dateOfBirth rather than the actual names of the fields (e.g., d.forename). Although using AS is not required, it can be considered good practise as it allows for better understanding of the query output as it gives a description of the data which has been produced.

```
MATCH (d:Driver) RETURN d.forename AS forename, d.surname AS surname,
d.dob AS dateOfBirth
```

The output of running this query can be found in figure 1 and the queries have been run using the Neo4j command-line tool. It is also possible to run the queries using the graphical user interface (GUI), although the command-line is a faster option, especially when needing to connect to a remote database via the university server due to the large size of the database being queried. The visualisation of the database is discussed in the ‘Neo4j Data Visualisation’ section of this chapter. The command

line can be started by running the command to start the tool, connecting to the database and then running the queries, and the setup is very quick and easy. The query returns the forenames, surnames and the date of births of all drivers as expected in a format which can be considered similar to SQL output. The Neo4j visualisation tool can be a very useful tool for data analysis, as it is simple to use, and the colours and labels can be fully customised, and the nodes and relationships can be dragged around to allow you to have a better overview of the data. The GUI is definitely a useful tool for the basic visualisation which will be explored further in this report. However, for large datasets it can take a very long time for the graphical instances of nodes and relationships to be loaded in (although returning large volumes of data in the command line is very quick) and can often crash due to the large amount of data being returned.



forename	surname	dateOfBirth
"Lewis"	"Hamilton"	"1985-01-07"
"Nick"	"Heidfeld"	"1977-05-10"
"Nico"	"Rosberg"	"1985-06-27"
"Fernando"	"Alonso"	"1981-07-29"
"Heikki"	"Kovalainen"	"1981-10-19"
"Kazuki"	"Nakajima"	"1985-01-11"
"Sébastien"	"Bourdais"	"1979-02-28"
"Kimi"	"Räikkönen"	"1979-10-17"
"Robert"	"Kubica"	"1984-12-07"
"Timo"	"Glock"	"1982-03-18"
"Takuma"	"Sato"	"1977-01-28"
"Nelson"	"Piquet Jr."	"1985-07-25"
"Felipe"	"Massa"	"1981-04-25"
"David"	"Coulthard"	"1971-03-27"
"Jarno"	"Trulli"	"1974-07-13"
"Adrian"	"Sutil"	"1983-01-11"
"Mark"	"Webber"	"1976-08-27"
"Jenson"	"Button"	"1980-01-19"
"Anthony"	"Davidson"	"1979-04-18"
"Sebastian"	"Vettel"	"1987-07-03"
"Giancarlo"	"Fisichella"	"1973-01-14"

Figure 1: Extract of Query Output Returning All Drivers

Test Query 2 – Return All British Drivers

This query returns the full names and date of births of all British drivers to have ever raced in formula one. The output to this query can be found in appendix G. This query is slightly more complex than the previous query, as it uses the WHERE keyword to filter the results to only display drivers who have a nationality of British. The output of this query is very similar to the query which returns all

drivers but returns less results as it only returns drivers who are British. Neo4j runs this very quickly and there were no issues which arose from this query.

```
MATCH (d:Driver) WHERE d.nationality = "British" RETURN d.forename AS
forename, d.surname AS surname, d.dob AS dateOfBirth
```

Test Query 3 – Return Lewis Hamilton's Pitstops

This query returns the total number of pitstops that Lewis Hamilton has completed in every British Grand Prix that he has ever raced in. This query is more complex than the other two test queries. The MATCH keyword states that this query will use the data related to the drivers, pitstops and the races which have been completed in F1. The WHERE keyword sets the conditions of the query: the first name of the driver must be 'Lewis' and the driver's surname 'Hamilton'. The race must be the British Grand Prix and join commands are used to link the drivers, pitstops and races together by their 'id' fields. In the RETURN statement there is a COUNT command which adds up all of the pit stops made by that driver at that specific year's race.

```
MATCH (d:Driver), (pt:pitStops), (r:race)
WHERE d.forename = "Lewis" AND r.name = "British Grand Prix" AND
d.driverId = pt.driverId AND r.raceId = pt.raceId
RETURN d.forename AS forename, d.surname AS surname, d.dob AS
dateOfBirth, count(pt.stop), r.name, r.year
```

The output of this query can be found in figure 2. It contains the driver's (only Lewis Hamilton is queried) full name and date of birth, as well as the race (only the British GP is queried) name and the year the race was held. The important column is the number of pitstops which were made by the driver (count.pt.stop) which shows how many pitstops Lewis Hamilton has made at each British GP he has raced in. This query ran very quickly and Neo4j coped with the query with no issues at all.

forename	surname	dateOfBirth	count(pt.stop)	r.name	r.year
"Lewis"	"Hamilton"	"1985-01-07"	3	"British Grand Prix"	"2011"
"Lewis"	"Hamilton"	"1985-01-07"	2	"British Grand Prix"	"2012"
"Lewis"	"Hamilton"	"1985-01-07"	2	"British Grand Prix"	"2013"
"Lewis"	"Hamilton"	"1985-01-07"	2	"British Grand Prix"	"2014"
"Lewis"	"Hamilton"	"1985-01-07"	2	"British Grand Prix"	"2015"
"Lewis"	"Hamilton"	"1985-01-07"	2	"British Grand Prix"	"2016"
"Lewis"	"Hamilton"	"1985-01-07"	1	"British Grand Prix"	"2017"
"Lewis"	"Hamilton"	"1985-01-07"	1	"British Grand Prix"	"2018"
"Lewis"	"Hamilton"	"1985-01-07"	1	"British Grand Prix"	"2019"

Figure 2: Query Output of Lewis Hamilton's Pit Stops

These queries all returned the values that were expected of them and therefore it can be said, with relative certainty that the data has been loaded correctly. Although it could have been better to show the data visualisation, it was not feasible due to the large number of results generated from this query which would lead to poor, unresponsive behaviour from the Neo4j GUI. The command line tool is an excellent option for querying data and allows for very quick query analysis to find out some interesting insights from the data. Occasionally, when querying a large number of columns, the output can become slightly untidy and difficult to read, although this is easily fixed by simply zooming out until the formatting is fixed. This does make the font size smaller, and therefore more difficult to read, but this will usually only be an issue in extreme circumstances.

Query Set 1

The queries within this set are called point queries, which are often considered the simplest category of queries, as they are used to locate a single driver. For example, they can be used to find drivers born in a specific year or find drivers who have completed exactly 25 races.

What is the optimal pitstop strategy of a race?

This query returns the number of the lap(s) that the pitstops of the winning driver happened on, which can be used to see if it is better to stop earlier or later during a race. First, it uses the data related to the drivers, pitstops, results and the races which have been held before filtering these results to only return the race winner as this shows that their pit strategy must have been a success. The rest of the

WHERE clause joins the driver, pitstops, results and races nodes together using their respective unique id fields. The query then returns the full names of the drivers, the race that took place and which year, and most importantly the lap number that the pit stop occurred. The DISTINCT keyword ensures that no duplicates were returned from this query. From exploring the output of this query, it seems like most race winners choose to stop earlier on if possible, which makes sense as it allows them to put on their chosen set of tyres as soon as possible, as drivers starting in the top 10 must start the race on the same set of tyres that they qualified on. Neo4j processed this query very quickly and there were no issues.

```
// return lap number of pitstop where the final race position
is first
MATCH (d:Driver), (pt:pitStops), (rr:results), (r:race)
WHERE d.driverId = pt.driverId AND d.driverId = rr.driverId
AND r.raceId = pt.raceId AND rr.raceId = pt.raceId AND
rr.position = "1"
RETURN DISTINCT d.forename AS forename, d.surname AS surname,
pt.lap, r.name, r.year
```

The output to this query can be found in Appendix I.

What is the ideal number of pitstops for a driver during a race?

A very similar query to the previous one, using the same structure and MATCH AND WHERE commands. The only difference is that within the RETURN statement, rather than finding the lap number of the pitstops, it returns a COUNT of the number of pitstops which the eventual race winner made during the race. This would provide a useful insight into whether it would be better to make fewer pitstops but have older tyres for much of the race or make more pitstops (thus spending more time in the speed restricted pitlane) but being able to have fresher tyres which would allow you to push the car more, especially in the latter stages of the race.

```
// return total number of pitstops where final race position
is first
MATCH (d:Driver), (pt:pitStops), (rr:results), (r:race)
```

```
WHERE d.driverId = pt.driverId AND d.driverId = rr.driverId
AND r.raceId = pt.raceId AND rr.raceId = pt.raceId AND
rr.position = "1"
RETURN d.forename AS forename, d.surname AS surname,
count(pt.stop), r.name, r.year
```

The output of this query (a sample of) is shown in the figure 3. It displays the driver, and the race which was held as well as the year it took place. Crucially, it shows the number of pitstops and it appears that most winners choose to stop the minimal number of times possible to reduce the amount of time spent in the pitlane. This may mean that they need to conserve their tyres in the latter stages of the race, but race winners will usually have the ability to be able to do this whilst still keeping at a good pace to ensure that they remain in front. The processing of this query was very quick with no issues occurring.

forename	surname	count(pt.stop)	r.name	r.year
"Sebastian"	"Vettel"	2	"Australian Grand Prix"	"2011"
"Sebastian"	"Vettel"	3	"Malaysian Grand Prix"	"2011"
"Lewis"	"Hamilton"	3	"Chinese Grand Prix"	"2011"
"Sebastian"	"Vettel"	4	"Turkish Grand Prix"	"2011"
"Sebastian"	"Vettel"	4	"Spanish Grand Prix"	"2011"
"Sebastian"	"Vettel"	1	"Monaco Grand Prix"	"2011"
"Jenson"	"Button"	6	"Canadian Grand Prix"	"2011"
"Sebastian"	"Vettel"	3	"European Grand Prix"	"2011"
"Fernando"	"Alonso"	3	"British Grand Prix"	"2011"
"Lewis"	"Hamilton"	3	"German Grand Prix"	"2011"
"Jenson"	"Button"	3	"Hungarian Grand Prix"	"2011"
"Sebastian"	"Vettel"	3	"Belgian Grand Prix"	"2011"
"Sebastian"	"Vettel"	2	"Italian Grand Prix"	"2011"
"Sebastian"	"Vettel"	3	"Singapore Grand Prix"	"2011"
"Jenson"	"Button"	3	"Japanese Grand Prix"	"2011"
"Sebastian"	"Vettel"	2	"Korean Grand Prix"	"2011"
"Sebastian"	"Vettel"	2	"Indian Grand Prix"	"2011"
"Lewis"	"Hamilton"	2	"Abu Dhabi Grand Prix"	"2011"
"Mark"	"Webber"	3	"Brazilian Grand Prix"	"2011"

Figure 3: Extract of Query Output Returning Number of Pit Stops of Race Winners

Are quicker pitstops more likely to lead to higher race positions? Or is consistency more important? Looking at the pitstops once again, this query aims to look at the finishing positions of drivers who completed some of the quickest pitstops. The MATCH keyword specifies that the drivers, pitstops and results data will be used in this query, and the WHERE clause asks for pitstops which took less than 17 seconds (a very quick time spent in the pitlane) and joins the data together using its respective id fields. It then RETURNs the names of the drivers, their final race position, the lap number of the

pitstop and the total amount of time spent in the pitlane in milliseconds. The ORDER BY keyword allows you to rearrange the data in ascending (or descending) order to make it easier to analyse certain trends of the dataset. This query orders the time of the pitstop in ascending order.

```
// returns all pitstops less than 17 seconds
MATCH (d:Driver), (pt:pitStops), (rr:results)
WHERE pt.milliseconds < "17000" AND d.driverId = pt.driverId
AND d.driverId = rr.driverId
RETURN DISTINCT d.forename AS forename, d.surname AS surname,
pt.milliseconds AS StopTime, pt.lap AS pittedOn, rr.position
AS racePosition
ORDER BY pt.milliseconds
```

A sample of the output of this query can be found in figure 4. As it turns out, most of the extremely quick pitstops only lead to fairly average finishing positions in the race. This may be because midfield times focus a lot of their time on getting as quick a pitstop as possible to try to get people talking about them in order to gain some good publicity for the team. Larger teams on the other hand, prioritise the need for consistency rather than outright speed, as having one good pitstop is not very helpful if your other pitstops are much slower which could cost you much needed track position. Just like the other queries tested so far, Neo4j copes with it with no issues and it is processed quickly.

forename	surname	StopTime	pittedOn	racePosition
"Jolyon"	"Palmer"	"1000234"	"9"	"11"
"Jolyon"	"Palmer"	"1000234"	"9"	"\N"
"Jolyon"	"Palmer"	"1000234"	"9"	"22"
"Jolyon"	"Palmer"	"1000234"	"9"	"13"
"Jolyon"	"Palmer"	"1000234"	"9"	"15"
"Jolyon"	"Palmer"	"1000234"	"9"	"12"
"Jolyon"	"Palmer"	"1000234"	"9"	"19"
"Jolyon"	"Palmer"	"1000234"	"9"	"10"
"Jolyon"	"Palmer"	"1000234"	"9"	"14"
"Jolyon"	"Palmer"	"1000234"	"9"	"17"
"Jolyon"	"Palmer"	"1000234"	"9"	"6"
"Sergio"	"Pérez"	"1000284"	"9"	"\N"
"Sergio"	"Pérez"	"1000284"	"9"	"17"
"Sergio"	"Pérez"	"1000284"	"9"	"14"
"Sergio"	"Pérez"	"1000284"	"9"	"9"
"Sergio"	"Pérez"	"1000284"	"9"	"11"
"Sergio"	"Pérez"	"1000284"	"9"	"7"
"Sergio"	"Pérez"	"1000284"	"9"	"15"
"Sergio"	"Pérez"	"1000284"	"9"	"10"

Figure 4: Extract of Query Output Returning Driver's Finishing Positions of Fastest Pitstops

Query Set 2

All of these queries are aggregation queries, which allow you to be able to derive group data by analysing individual records, such as looking at the total number of wins by a certain driver or finding the average number of points achieved each season by a team.

Which country has the most championship points?

This query takes all of the countries which have had drivers representing them and looks at which country has the most points. It takes the data relating to the drivers and the race results (the MATCH clause) and joins them using their unique id fields. The query then returns the nationalities of all of the drivers and adds up all of the points gained by the drivers of these countries. The ORDER BY keyword sorts the number of championship points in ascending order.

```
// return total points for all drivers, with nationality to
see which country is most successful
MATCH (d:Driver), (rr:results)
WHERE d.driverId = rr.driverId
RETURN d.nationality, sum(toInteger(rr.points))
ORDER BY sum(toInteger(rr.points))
```

Figure 5 shows the output of this query. As can be seen from the output, there are a number of countries which have had drivers who have never achieved a point in Formula One, such as Malaysia and Czechia. As expected, Britain leads the way in terms of points as early on in the sport's history most of the races were in or near the UK, and the vast majority of drivers were from Britain. Germany is relatively close behind, helped hugely by the dominance of Michael Schumacher, with Finland (Kimi Raikkonen and Mika Hakkinen), Brazil (Ayrton Senna and Rubens Barrichello) and France (Alain Prost) rounding off the top five. This query was processed very quickly by Neo4j with no issues being recorded.

d.nationality	sum(toInteger(rr.points))
"Malaysian"	0
"Czech"	0
"Liechtensteiner"	0
"American-Italian"	0
"Uruguayan"	0
"Argentine-Italian"	0
"East German"	0
"Indonesian"	0
"Hungarian"	1
"Chilean"	3
"Indian"	5
"Rhodesian"	6
"Portuguese"	8
"Irish"	16
"Venezuelan"	77
"Thai"	100
"Danish"	158
"Japanese"	209
"Russian"	235
"Polish"	274
"South African"	282
"Colombian"	307
"Monegasque"	307
"Swiss"	348
"Belgian"	388
"Swedish"	396
"Canadian"	409
"New Zealander"	541
"Mexican"	675
"Argentine"	696
"Dutch"	975
"Austrian"	990
"American"	996
"Italian"	2029
"Spanish"	2248
"Australian"	2561
"French"	3027
"Brazilian"	3422
"Finnish"	3875
"German"	7810
"British"	8946

Figure 5: Query Output of Total Championship Points for each Country

Which driver would win an overall World Championship since 1950?

This query takes the data related to the F1 drivers and the results of each of the drivers at every race, and season as a whole. It converts the number of wins of each driver to the integer data type and counts them to provide the total number of wins of each driver, which is then displayed to the user. As with the previous queries, the driver and driverStandings nodes are joined together by their respective unique id fields. The full names of the drivers and their total number of race wins are then returned.

```
// return the total number of wins of each driver
MATCH (d:Driver), (ds:driverStandings)
WHERE d.driverId = ds.driverId  AND toInteger(ds.wins)  >= 1 AND
ds.position = "1"
RETURN DISTINCT d.forename AS forename, d.surname AS surname,
count(toInteger(ds.wins)) AS totalWins
```

The output of the query can be found in figure 6. It contains the first names and surnames of the drivers, followed by the total number of wins achieved by that driver. An example is that Lewis Hamilton has achieved 94 wins within the sport. This was actually one of the more difficult queries to complete as it kept returning incorrect values, either by counting duplicate values or not counting enough, which was unexpected when the query was first thought of. Eventually, by using the DISTINCT keyword and the relevant join commands the query allowed for the correct output to be produced and the query was processed with no issues from Neo4j.

forename	surname	totalWins
"Kimi"	"Räikkönen"	3
"Lewis"	"Hamilton"	94
"Felipe"	"Massa"	1
"Fernando"	"Alonso"	43
"Michael"	"Schumacher"	111
"Mika"	"Häkkinen"	26
"Eddie"	"Irvine"	3
"Jacques"	"Villeneuve"	5
"Damon"	"Hill"	16
"Ayrton"	"Senna"	41
"Alain"	"Prost"	66
"Nigel"	"Mansell"	22
"Jenson"	"Button"	18
"Nelson"	"Piquet"	12
"Michele"	"Alboreto"	1
"Niki"	"Lauda"	37
"John"	"Watson"	3
"Didier"	"Pironi"	4
"Carlos"	"Reutemann"	10
"René"	"Arnoux"	3
"Alan"	"Jones"	7
"Jacques"	"Laffite"	3
"Gilles"	"Villeneuve"	2
"Jody"	"Scheckter"	12
"Mario"	"Andretti"	11
"James"	"Hunt"	1
"Emerson"	"Fittipaldi"	20
"Jackie"	"Stewart"	27
"Jochen"	"Rindt"	8
"Graham"	"Hill"	15
"Denny"	"Hulme"	5
"Jack"	"Brabham"	16
"Jim"	"Clark"	19
"John"	"Surtees"	1
"Wolfgang"	"von Trips"	2
"Phil"	"Hill"	2
"Stirling"	"Moss"	4
"Juan"	"Fangio"	27
"Peter"	"Collins"	1
"Alberto"	"Ascari"	12
"Nino"	"Farina"	3
"Mark"	"Webber"	6
"Sebastian"	"Vettel"	55
"Nico"	"Rosberg"	24
"Valtteri"	"Bottas"	1

Figure 6: Query Output of Total Wins Achieved by each Driver

Do drivers qualifying on pole position usually win the race?

This query links together the drivers, race results and qualifying data to try to find out the average qualifying and race positions for each driver. For this query, the WHERE clause simply joins the data together using their respective unique id fields. The full names of the drivers are returned, along with the average qualifying and race positions of each driver. These values are calculated by converting the values to the float data type, then using the built-in function (avg) to calculate the average of all of the

drivers' positions. Finally, the query sorts the driver's average finishing position in the race into ascending order.

```
// return the average qualifying, and race, positions of every
driver
MATCH (d:Driver), (rr:results), (qt:qualifying)
WHERE d.driverId = qt.driverId AND d.driverId = rr.driverId
RETURN d.forename AS forename, d.surname AS surname,
avg(toFloat(qt.position)) AS qualifyingPosition,
avg(toFloat(rr.position)) AS racePosition
ORDER BY avg(toFloat(rr.position))
```

Figure 7 shows the output returned by this query. For the high average finish positions in races, the average qualifying position is usually quite similar, which could be as expected as the top drivers will usually be very consistent and be unlikely to make lots of mistakes which would result in them losing positions (and therefore affecting their average). In contrast, drivers finishing (on average) in the midfield positions often start the races lower down in the order, which is quite interesting. One possible reason for this is that the midfield is often very congested, especially at the start of a grand prix, which can often result in many incidents which could force drivers to retire, so therefore allowing drivers who get clear of the incidents to finish higher up in the standings which provides a boost to the average race position. Despite this query requiring some real-time calculations and type conversions, Neo4j coped with the query processing with no issues.

forename	surname	qualifyingPosition	racePosition
"Ayrton"	"Senna"	1.0	3.1481481481481475
"Lewis"	"Hamilton"	3.564000000000009	3.3628318584070525
"Michael"	"Schumacher"	5.666666666666713	3.7012448132780023
"Nigel"	"Mansell"	4.833333333333342	3.918367346938776
"Sebastian"	"Vettel"	4.641666666666614	3.9428571428571626
"Juan"	"Pablo Montoya"	6.03125	4.2096774193548425
"Mika"	"Häkkinen"	6.048387096774175	4.466019417475727
"Damon"	"Hill"	5.372881355932205	4.480519480519482
"Max"	"Verstappen"	6.666666666666671	5.073170731707295
"Gerhard"	"Berger"	5.760869565217371	5.096774193548384
"Kimi"	"Räikkönen"	6.545774647887317	5.3585657370518165
"David"	"Coulthard"	9.193939393939369	5.828402366863933
"Fernando"	"Alonso"	7.8787878787877945	5.8779527559055165
"Eddie"	"Irvine"	8.265624999999982	5.988235294117652
"Valtteri"	"Bottas"	6.3500000000000325	6.179687499999989
"Jean"	"Alesi"	7.467741935483896	6.210084033613442
"Ralf"	"Schumacher"	8.261682242990606	6.254098360655708
"Nico"	"Rosberg"	6.834951456310701	6.3559322033898455
"Heinz-Harald"	"Frentzen"	9.037037037037054	6.520833333333327
"Mark"	"Webber"	7.31683168316825	6.600000000000036
"Charles"	"Leclerc"	8.238095238095225	6.852941176470587
"Jacques"	"Villeneuve"	9.542168674699878	7.046296296296316
"Rubens"	"Barrichello"	9.2699115044248	7.056768558951996
"Martin"	"Brundle"	11.374999999999979	7.192771084337337
"Felipe"	"Massa"	8.357976653696525	7.266375545851552
"Jenson"	"Button"	9.334600760456151	7.489090909090966

Figure 7: Extract of Query Output Returning Average Qualifying and Race Positions for each Driver

Query Set 3

The first of these queries can be considered to be another aggregation query, as seen in Query Set 2, but the final two queries are range queries, which is where you aim to locate all of the records which have values between two specific boundaries.

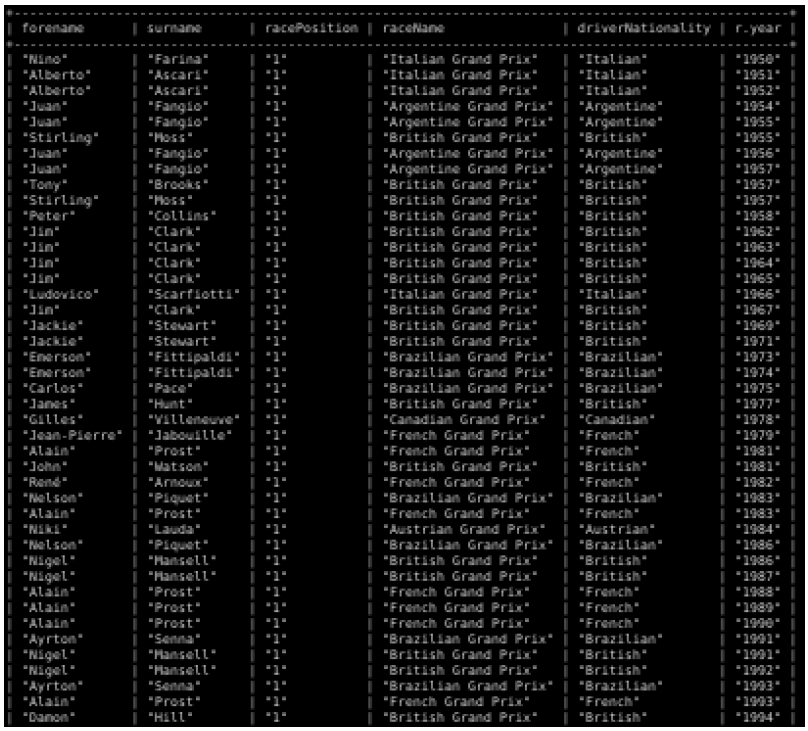
Do drivers usually win Grand Prix in their home country?

This query uses the data related to the F1 drivers, the race results, the circuits which have been used and the races that have been held, which are then joined together using their respective unique id fields. The WHERE clause uses the inbuilt Neo4j function 'left', which allows you to get a substring of the start of a data field, specified by a number depending on the number of characters you need (e.g., left("example", 2) would return "ex"). This allows you to compare the nationality of the driver with the name of the racetrack used but taking away the 'Grand Prix' of the race name so that the two strings can be compared. This query returns the full names of all of the drivers, the final race position (which will always be first), the name of the race and the nationality of the driver, as well as the year of the race which is sorted into ascending order.

```
// return the driver's nationality and the country where the
race was held if they are the same, where the finish position
was first
MATCH (d:Driver), (rr:results), (t:track), (r:race)
WHERE d.driverId = rr.driverId AND t.circuitId = r.circuitId
AND rr.raceId = r.raceId AND (d.nationality = left(r.name, 6)
OR d.nationality = left(r.name, 7) OR d.nationality =
left(r.name, 8) OR d.nationality = left(r.name, 9) OR
d.nationality = left(r.name, 10) OR d.nationality =
left(r.name, 11) OR d.nationality = left(r.name, 12)) AND
rr.position = "1"
RETURN DISTINCT d.forename AS forename, d.surname AS surname,
rr.position AS racePosition, r.name AS raceName, d.nationality
AS driverNationality, r.year
ORDER BY r.year
```

Figure 8 shows the output of the query. It shows the drivers who have won the race from the same country that they are from. Some examples are when Ayrton Senna, who is Brazilian, won the 1991 Brazilian Grand Prix and Lewis Hamilton (British) won the 2017 British Grand Prix. This query

shows that a lot of drivers have won a race from the country of the nationality. This could be due to a number of different factors, such as having the home supporters cheering you on and giving you some extra momentum to do well or because the drivers have a large amount of familiarity with the track as it is likely they have raced at the circuit earlier in their career. Neo4j processed the query absolutely fine and there were no issues when creating thus query,



forename	surname	racePosition	raceName	driverNationality	r.year
"Nino"	"Farina"	"1"	"Italian Grand Prix"	"Italian"	"1950"
"Alberto"	"Ascari"	"1"	"Italian Grand Prix"	"Italian"	"1951"
"Alberto"	"Ascari"	"1"	"Italian Grand Prix"	"Italian"	"1952"
"Juan"	"Fangio"	"1"	"Argentine Grand Prix"	"Argentine"	"1954"
"Juan"	"Fangio"	"1"	"Argentine Grand Prix"	"Argentine"	"1955"
"Stirling"	"Moss"	"1"	"British Grand Prix"	"British"	"1955"
"Juan"	"Fangio"	"1"	"Argentine Grand Prix"	"Argentine"	"1956"
"Juan"	"Fangio"	"1"	"Argentine Grand Prix"	"Argentine"	"1957"
"Tony"	"Brooks"	"1"	"British Grand Prix"	"British"	"1957"
"Stirling"	"Moss"	"1"	"British Grand Prix"	"British"	"1957"
"Peter"	"Collins"	"1"	"British Grand Prix"	"British"	"1958"
"Jim"	"Clark"	"1"	"British Grand Prix"	"British"	"1962"
"Jim"	"Clark"	"1"	"British Grand Prix"	"British"	"1963"
"Jim"	"Clark"	"1"	"British Grand Prix"	"British"	"1964"
"Jim"	"Clark"	"1"	"British Grand Prix"	"British"	"1965"
"Ludovico"	"Scarfiotti"	"1"	"Italian Grand Prix"	"Italian"	"1966"
"Jim"	"Clark"	"1"	"British Grand Prix"	"British"	"1967"
"Jackie"	"Stewart"	"1"	"British Grand Prix"	"British"	"1969"
"Jackie"	"Stewart"	"1"	"British Grand Prix"	"British"	"1971"
"Emerson"	"Fittipaldi"	"1"	"Brazilian Grand Prix"	"Brazilian"	"1973"
"Emerson"	"Fittipaldi"	"1"	"Brazilian Grand Prix"	"Brazilian"	"1974"
"Carlos"	"Pace"	"1"	"Brazilian Grand Prix"	"Brazilian"	"1975"
"James"	"Hunt"	"1"	"British Grand Prix"	"British"	"1977"
"Gilles"	"Villeneuve"	"1"	"Canadian Grand Prix"	"Canadian"	"1978"
"Jean-Pierre"	"Jabouille"	"1"	"French Grand Prix"	"French"	"1979"
"Alain"	"Prost"	"1"	"French Grand Prix"	"French"	"1981"
"John"	"Watson"	"1"	"British Grand Prix"	"British"	"1981"
"René"	"Arnoux"	"1"	"French Grand Prix"	"French"	"1982"
"Nelson"	"Piquet"	"1"	"Brazilian Grand Prix"	"Brazilian"	"1983"
"Alain"	"Prost"	"1"	"French Grand Prix"	"French"	"1983"
"Niki"	"Lauda"	"1"	"Austrian Grand Prix"	"Austrian"	"1984"
"Nelson"	"Piquet"	"1"	"Brazilian Grand Prix"	"Brazilian"	"1986"
"Nigel"	"Mansell"	"1"	"British Grand Prix"	"British"	"1986"
"Nigel"	"Mansell"	"1"	"British Grand Prix"	"British"	"1987"
"Alain"	"Prost"	"1"	"French Grand Prix"	"French"	"1988"
"Alain"	"Prost"	"1"	"French Grand Prix"	"French"	"1989"
"Alain"	"Prost"	"1"	"French Grand Prix"	"French"	"1990"
"Ayrton"	"Senna"	"1"	"Brazilian Grand Prix"	"Brazilian"	"1991"
"Nigel"	"Mansell"	"1"	"British Grand Prix"	"British"	"1991"
"Nigel"	"Mansell"	"1"	"British Grand Prix"	"British"	"1992"
"Ayrton"	"Senna"	"1"	"Brazilian Grand Prix"	"Brazilian"	"1993"
"Alain"	"Prost"	"1"	"French Grand Prix"	"French"	"1993"
"Damon"	"Hill"	"1"	"British Grand Prix"	"British"	"1994"

Figure 8: Extract of Query Output Returning Races Where Host Country is Same as the Winner's Nationality

What lap is the fastest lap of a race usually set?

This query is designed to show on which lap the fastest lap of the race was set. It uses the data related to the drivers, race results and the races which have been held in F1, which are then joined together using their respective unique id fields. The RETURN clause contains the full names of the drivers', the final race position, the lap number of the fastest lap, the time of the fastest lap and the name and year of the race which was held.

```
// return the fastest lap time of each driver, when it was
set, the final race position of the driver, and the race being
held
MATCH (d:Driver), (rr:results), (r:race)
WHERE d.driverId = rr.driverId AND rr.raceId = r.raceId
```

```
RETURN DISTINCT d.forename AS forename, d.surname AS surname,
rr.position AS racePosition, rr.fastestLap AS fastestLap,
rr.fastestLapTime AS fastestLapTime, r.name AS raceName,
r.year AS yearHeld
```

An extract of the output of this query can be found in Appendix H. It contains a list of all of the drivers and looks at the lap number of their fastest lap of the race. Looking at the data, it is clear to see that a large number of fastest laps were set during the latter stages of a race. This could be because a lot of positions have already been decided earlier in the race, and so drivers may decide to pit for a faster set of tyres (knowing that there is little risk of being overtaken whilst they are in the pitlane) to try to get a fastest lap. This strategy has been even more common recently, as a recent rule change gave drivers an extra championship points for getting the fastest lap, providing they still finished in the top 10 positions. The query processing was done by Neo4j quickly with no issues.

Do younger drivers win more races than older drivers?

The final query being used to analyse the Neo4j technology uses the data related to the drivers and the race results, which are joined together using their respective unique id fields. The RETURN clause contains the first names, surnames and date of births of the drivers, as well as the average race positions of the drivers (which had been converted into the float data type). The date of the births of the drivers' have been sorted into ascending order.

```
// return the date of birth and final race position of each
driver
MATCH (d:Driver), (rr:results)
WHERE d.driverId = rr.driverId
RETURN DISTINCT d.forename AS forename, d.surname AS surname,
d.dob AS dateOfBirth, avg(toFloat(rr.position)) AS
racePosition
ORDER BY d.dob
```

The output to this query can be found in Appendix J. The query shows the full names and date of births of the drivers as well as their average race positions. The query shows that some of the youngest drivers have very good highest average positions, which suggests that your age will not stop you from becoming a success in Formula One, and it is your ability and being given a good car to race

in which is far more important. Some examples are Max Verstappen (the youngest race winner in Formula One) has an average position of 5th and Charles Leclerc with an average position of 6th. This query shows that if you are skilled enough then you are probably old enough, although in F1 sometimes you need some luck to be given a car which is capable of producing the results you are capable of. As with all of the queries which have been run, Neo4j completed the query processing quickly and with no issues occurring.

Neo4j Data Visualisation

After some minor troubleshooting to work how it would be possible to view the database visualisation, it was eventually found that there was a way to view it on the Heriot Watt server. Although it might have been easier to view the visualisation on a local machine, due to different versions of Neo4j being used this was not possible and so using the server was the next best option. This meant that it was now possible to view how the graph database of the F1 dataset appeared and be able to use Neo4j in one of the main ways that it was intended to be used.

Although it would have been nicer to be able to use the Neo4j Bloom plugin to make more interesting visualisations, which could be further customised using some interesting features to help to better understand the dataset, the visualisation which Neo4j can produce on its own without any plugins can still be very insightful and allow people of differing technical expertise to view the dataset and understand what it being presented. It would be very interesting for future projects to explore the use of Neo4j Bloom and how much better it is at visualising data, as well as how useful it can be at presenting data to various stakeholders to help better understand a dataset.

In order to look at the use of Neo4j when analysing a dataset, a number of visualisations will be run to look at how effective the tool can be at analysing this particular set of data.

The first task when creating a visualisation of a dataset is to look at an overview of the data and how it interconnects with all of the different parts. Appendix C shows an initial dataset diagram which shows the nodes and relationships, and the visualisation within Neo4j provides something very similar (figure 9). The nodes of the data are represented by circles, and these can be customised with various colours to make it easy to understand. From this visualisation below some of the nodes have been put into different colours to show that they represent different parts of the dataset, although some of the nodes have not been changed and they are represented by a grey colour. This has been done to show that although you can choose to have all of the nodes the same colour, this is definitely not the recommended method as it can make it very confusing and overwhelming with the amount of data being represented with no clear differences about which parts represent which pieces of data. These nodes are joined together by lines which represent the relationships. An important point to note when doing the visualisation is that when there is a large number of relationships, it can easily quite cluttered, and it can be difficult to see which relationships lead to which nodes. This means that you will probably need to zoom out and rearrange all of the nodes and relationships to be able to view all of the at the same time in a clear and easy-to-understand way. This can take quite a long time to have the data presented in such a way that you can view all of the nodes and relationships easily, but once it has been done it can provide some very useful insights into how the data is connected. Seeing the connections within the data can be useful in helping to construct queries which allows you to further analyse the dataset in ways that are very difficult to do using data visualisation techniques.

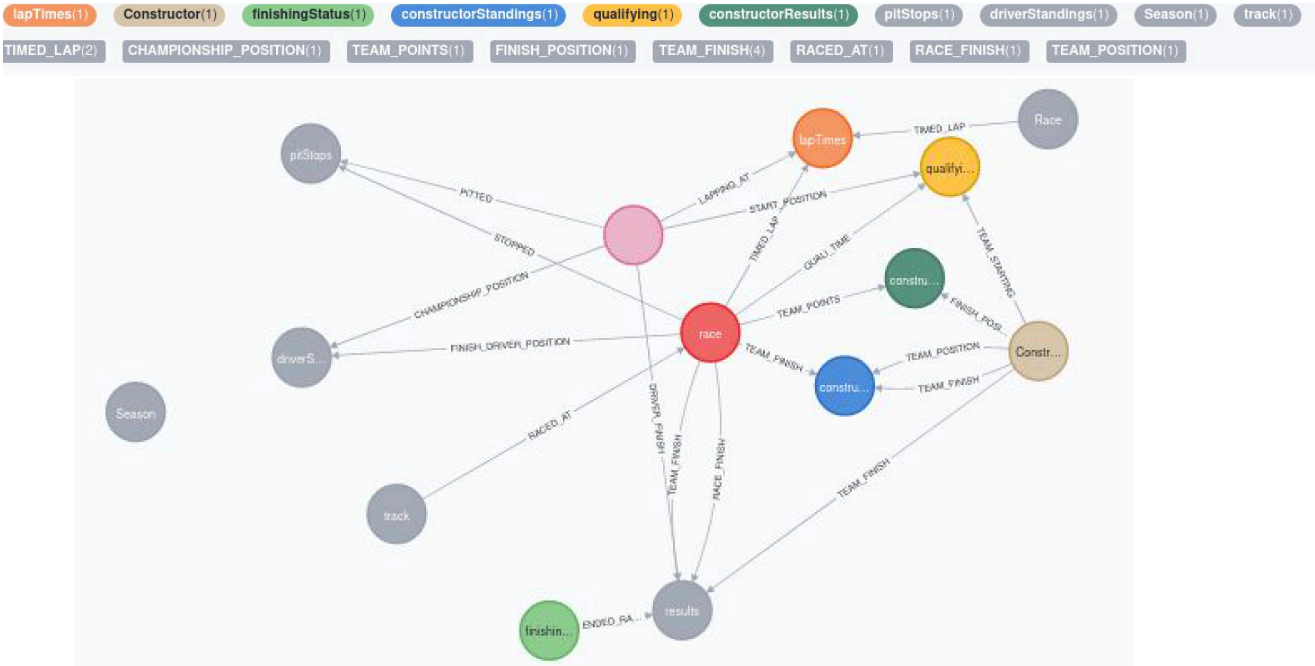


Figure 9: Neo4j Schema Visualisation of the F1 Dataset

Figure 10 shows a visualisation of all of the drivers who have ever raced in Formula One, which is a total of 848. An important thing to remember when doing data visualisations is Neo4j is that it is most valuable when you can see the relationships and how various aspects of the data are interconnected. This visualisation which only shows the nodes of the drivers is not particularly useful, as it does not tell us anything about how the data is related each other. However, it does still have a benefit over the command line as it looks much nicer, and you can easily click on one of the nodes to see further details about the driver you selected.

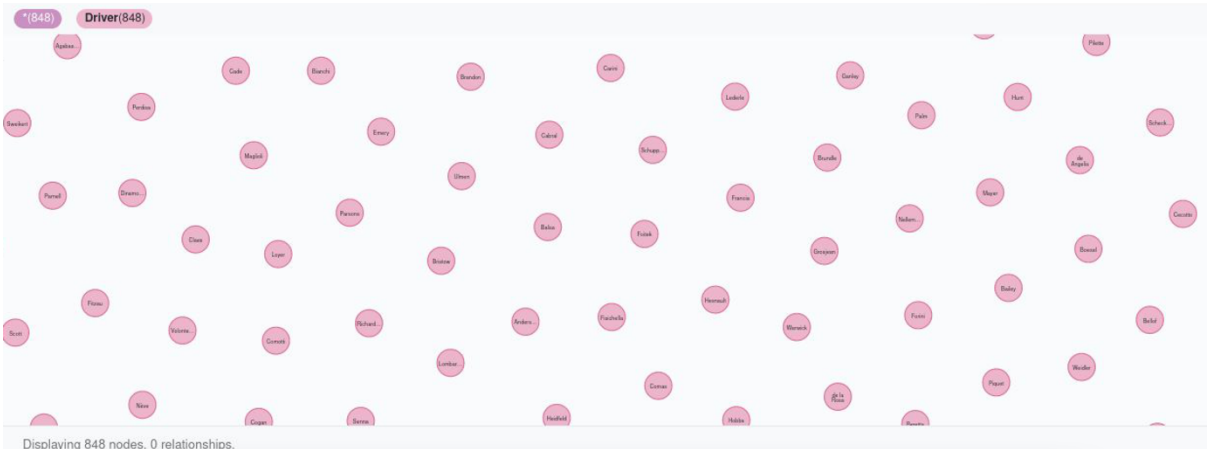


Figure 10: Visualisation Showing All Drivers

Figure 11 shows one driver (Lewis Hamilton – pink node) and a list of all his championship positions (grey nodes), which are linked together as relationships. This visualisation is more interesting, as you can now begin to see how all of the data is interconnected with each other. As you move around the various nodes you can see the lines which are linked together and how the data has been stored in the database. Of course, you can choose to customise the colours to better suit the dataset, or even just to make it more interesting to look at, and further details about the dataset can be obtained by clicking on that specific area of the graph database.

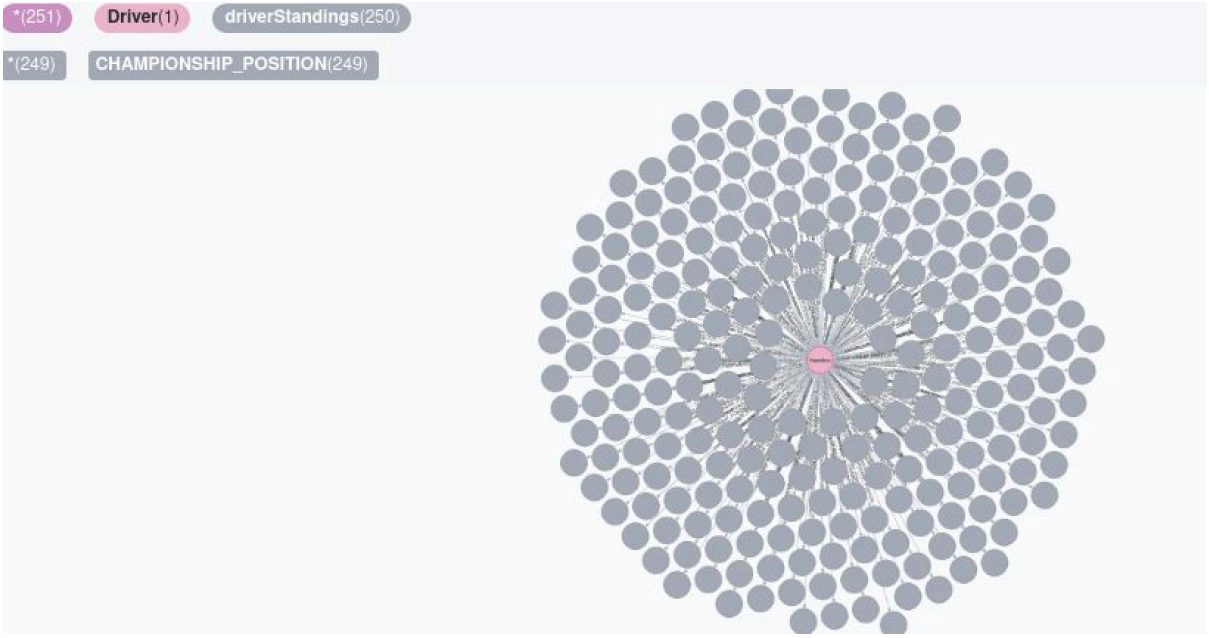


Figure 11: Visualisation Showing All Championship Positions for Lewis Hamilton

The next visualisation is very similar, but rather than showing the championship positions for just one driver, it shows these details for every driver to have raced in Formula One (figure 12). This provides some insightful data, as by simply looking around the dataset you can easily see which drivers have participated in lots of races, and which have only taken part in a small number. For example, the driver on the far right can clearly be seen to not have taken part in many races, which can lead you to further insights such as it is unlikely that he would have won the Driver's Championship with such a small number of races competed in, which could then be explored further. A driver with a large number of grey nodes linked to them is much more experienced and there is a higher chance of having

a world title amongst their honours and will probably have more championship points which they have obtained, although this is not necessarily true as perhaps, they spent their long F1 career in one of the worst cars at the back of the grid struggling to get any points finishes. This is why it is important that you do not just rely on initial visualisations, and that you continue to explore the dataset as much as possible to find out as much information as you can to find some insights you may not have even thought of. An important thing to note when doing data visualisations with Neo4j is that with a large number of nodes and relationships it can often slow down quite substantially and on some occasions even end up crashing. This is why this screenshot was limited to a total of 5000 nodes to prevent slowdown of the machine and it would be interesting, in a future project, to see if the Neo4j Bloom plugin was able to better handle large quantities of data.

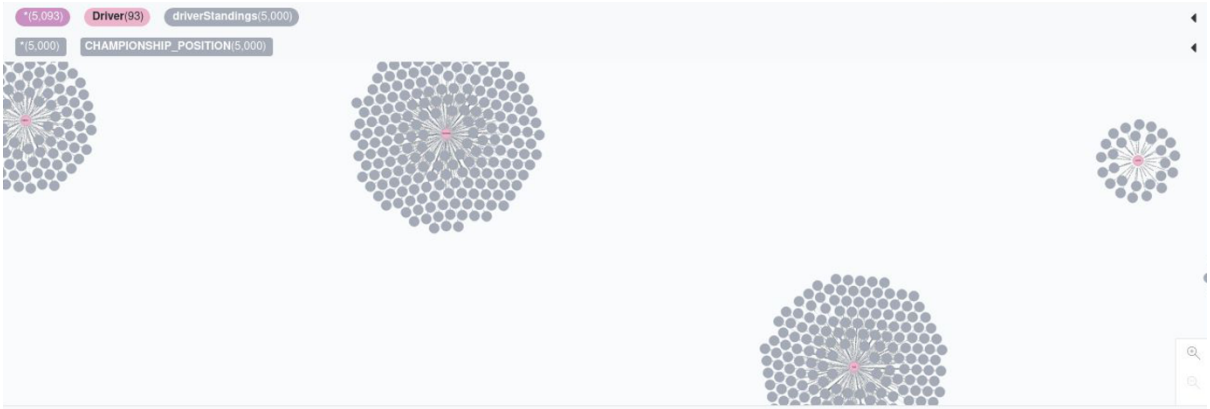


Figure 12: Visualisation Showing Championship Positions for All Drivers

Figure 13 shows all of the circuits (e.g., Silverstone – grey nodes) which have held races (e.g., 2018 British GP) in Formula One. This visualisation shows useful information as it can be seen which tracks have been on the race calendar for the longest amounts of time, which might suggest that they have some characteristics (such as long, sweeping fast corners) which are well suited to the sport. This could be useful if you were planning to create a new circuit and you wanted to see which sections worked well and which you might be able to incorporate into your own design. A lack of races held could also mean that there were some political issues which prevented them (e.g., the Bahrain GP was cancelled for a few years due to rising tensions in the region) or perhaps there was a lack of funding available, as it is very expensive to host a F1 race, as each country must pay an often-

substantial hosting fee to the competition organisers every year. An example within this screenshot (on the left-hand side) is the Aintree Grand Prix, which held the British race 5 times between 1955 and 1962 (alternating with Silverstone every 2 years). In 1957 it brought 150,000 spectators to watch the race but very quickly afterwards it was decided that it would no longer be a F1 host, probably due to rising costs, as half of the track was soon converted to be used as a horse racing track. The colours used in the visualisation, which can be fully customised, make it much more interesting to look at and analyse than the more traditional relational databases such as SQL.

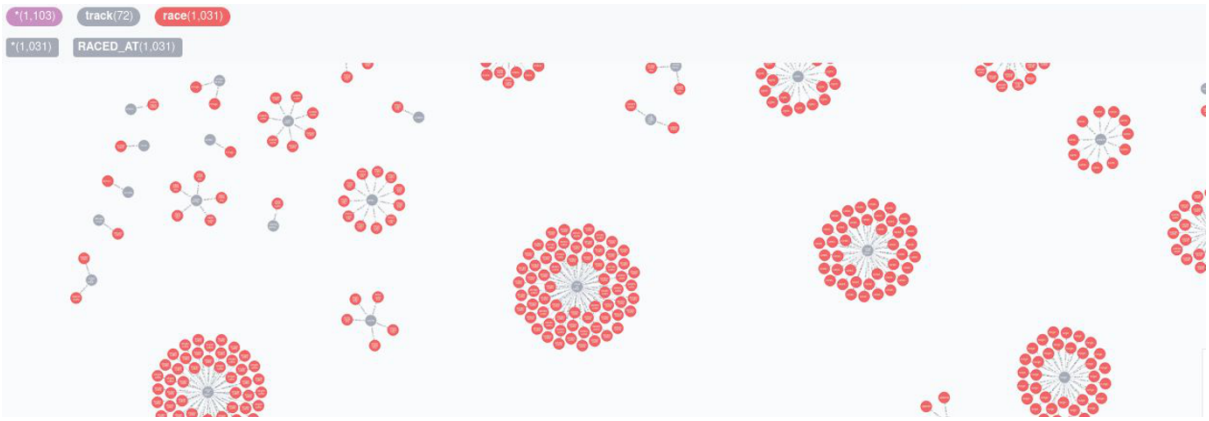


Figure 13: Visualisation Showing All Races Held by each Circuit

The next data visualisation shows the qualifying times (yellow nodes) which have been set at the different F1 races (red nodes). Figure 14 shows a number of races which have not been used often in Formula One, as there is only a small number of qualifying times being displayed which have been set at these races. This could simply be because they are relatively new tracks, as because of the COVID-19 global pandemic the 2020 F1 season had to be quickly rearranged using many circuits which have never been used in the sport, which had minimal travel restrictions so that teams could travel to the countries, so that a season could take place at all in the circumstances. This meant that there was a number of circuits which were awarded a race which would never normally get the chance to host a F1 event which means that they only have the qualifying times for that one season. The other reason of only having a small number of qualifying times is because the circuits only held races early on in F1 history (such as in the 1950s), where only a small number of drivers took part, partly due to the high costs involved in building a capable car. When scrolling through the visualisation, it is easy to see the circuits which have been on the race calendar for the longest, as they are surrounded by a large

number of yellow nodes (qualifying times which have been set) and it provides a useful insight into the costs involved with hosting a Formula One event.

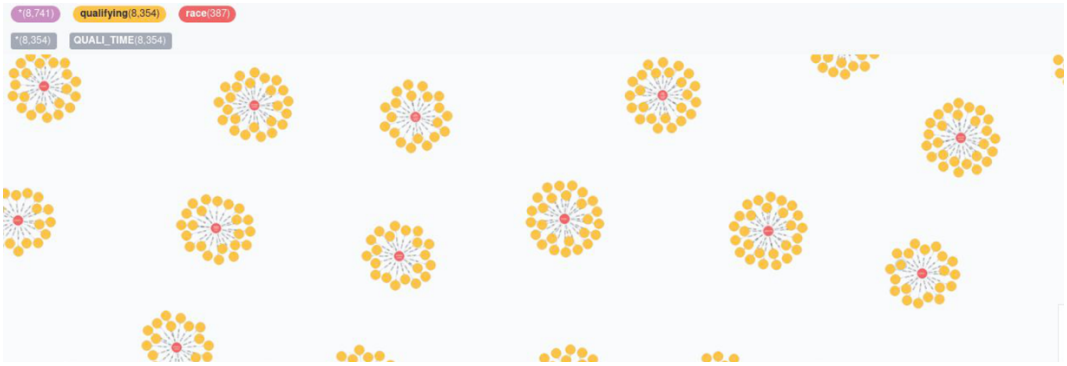


Figure 14: Visualisation Showing the Qualifying Times of each F1 Race

The final visualisation (figure 15) shows the number of pitstops (grey nodes) which have been completed by each driver (pink nodes). Usually, drivers complete on average between 1-2 pitstops per race, which means that this information can be used to estimate how many races they have completed at a glance at the visualisation, but also how many seasons they have completed in the sport. For example, the two drivers on the left of the screenshot have most likely only completed one season in F1, which could because of poor performance (likely if placed in a high-pressure environment of a large team such as Red Bull Racing) or perhaps due to a lack of sponsorship opportunities (more likely if part of a smaller teams such as Haas which relies on driver sponsors to survive every season). However, making estimates on the number of pitstops is not always entirely accurate, as sometimes there are tyres of less durability available one season which means that more pitstops need to be made, or when refuelling during the race was permitted which often led to drastic different strategies which resulted in a wide range in the number of pitstops completed between drivers.

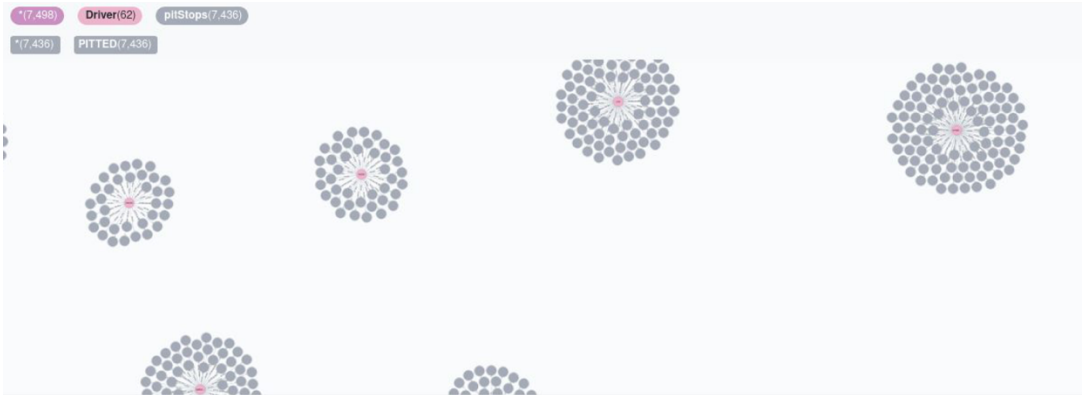


Figure 15: Visualisation Showing Number of Pitstops Completed by each Driver

Project Assessment

This section of the document details how effectively the work has been carried out throughout the project, and if anything did not go as expected. In order to ensure that the best work possible has been produced, a project timetable was created to ensure that everything was completed on time and to a high standard.

Project Evaluation

In general, this project can be considered a success as it was able to analyse the use of the graph database modelling tool Neo4j in analysing a Formula One dataset. The entire dataset was loaded in successfully, although one of the csv files was slightly troublesome as it took a very long time to load in. However, this allowed for further research to be made into the topic and it showed the importance of using indexes in Neo4j as it substantially reduced the amount of time taken to load in the data into the database. Interestingly, if data loading is a major weakness of Neo4j, one of the key strengths of the tool is the analysis of queries, as it was able to complete all of the queries very quickly. As well as the speed of the query processing, it is very easy to learn if you have any experience of traditional relational databases as it uses a very similar structure to SQL. Being able to visualise the data, rather than solely using the command line like traditional alternatives is a major selling point of graph databases as it allows many different people within a team, of various levels of technical expertise, to explore the dataset and how it is structured. In the future, it would have been better (and more convenient) if the database could be stored on a local machine rather than needing to be stored remotely on a university server. However, this requires a more powerful machine which is expensive and most of the required tasks could be completed anyway. Having the opportunity in the future to explore the Neo4j Bloom plugin could provide some very intriguing insight into the use of database visualisation and allow for more research into this area rather than solely looking at the more basic visualisation package of the base Neo4j software.

Although it was not the primary purpose of this study, my background of using Structured Query Language (SQL) has allowed me to look at how it compares to Neo4j. As someone who has experience using SQL, it was very easy to learn how to use Neo4j and be able to run the commands needed to analyse the dataset. However, it did take some time to learn about some additional details of the software, such as how important the use of indexes is at loading in the csv files. Although Neo4j first appears to be very slow at loading in the data (Sharma et al, 2018), with a small amount of research it became clear that by adding a small number of indexes it would reduce the loading time substantially. Neo4j's dependence on the use of queries to speed up the data loading does require you to think about which fields would most benefit from them, as indexes do take up storage space, but after some experience with using Neo4j this task gets easier and eventually you will get quite good at knowing where indexes would be most beneficial.

On the other hand, I believe that Neo4j is much better at running and processing queries than SQL. Initially, even constructing the queries was much easier than SQL as you do not need to use numerous JOIN commands which often results in the SQL queries being very long in length and can often get very confusing very quickly (Fernandes and Bernardino, 2018), especially when needing to troubleshoot after encountering an unexpected problem. From my experience throughout this project, it appears that Neo4j is at least as quick as SQL at being able to process queries, but, in my opinion, the main benefit is the visualisation of the query results. Being able to look at graph of the query to find out some interesting insights is much more interesting than looking at tables on the command line and being able to customise these visualisations in many different ways allows you to present the information in the best way for what you need it for and allows many other people to understand what is being presented as it is shown in an easy-to-understand format.

In general, I believe that the open-source software of Neo4j has shown lots of promise throughout this project and I think that graph-based data technologies are the way forward for data exploration. If the Neo4j team can continue to work on improving the speed of their software, especially when loading in the data, to at least match (or be even faster) than traditional alternatives like SQL then it will not

be long before Neo4j increases exponentially in popularity and many companies end up switching to graph-based systems. The ease of use, and the ability to visualise and customise the data in a way which suits you and your team cannot be underestimated and Neo4j provides a method of visualising data which traditional methods simply cannot match.

Project Timetable

The project timetable allowed for a good overview of the plan and overall structure of the project and ensured that progress was continually being made at all stages. Although the data insertion stage took slightly longer than intended, mainly due to the size of the lap_times.csv file and underestimating how slow Neo4j is at data loading, the queries were done quite quickly and actually took less time than was initially delegated. Splitting up the queries to be done into multiple iterations proved to be a good idea, as it allowed for the analysis to be carried out in smaller, more manageable sections and the knowledge gained in the earlier iterations would prove to be beneficial when solving the queries in the second and third iterations. The dissertation document was completed whilst the work was being carried out and therefore it did not leave lots of work to do at the end of the project.

Description of Task	**Date to be Started**	**Date to be Completed**	**Date Actually Completed**
Research of the Software Required	11 December 2020	21 December 2020	21 December 2020
Data insertion	28 December 2020	6 January 2021	12 February 2021
ITERATION 1: SAMPLE QUERIES 1, 2, 3			
Run the queries for first iteration	7 January 2021	12 January 2021	21 February 2021
Analyse the queries for first iteration	14 January 2021	29 January 2021	25 February 2021
Analyse the technology	1 February 2021	5 February 2021	28 February 2021
ITERATION 2:			

SAMPLE QUERIES 4, 5, 6			
Run the queries for second iteration	8 February 2021	11 February 2021	2 March 2021
Analyse the queries for second iteration	12 February 2021	19 February 2021	8 March 2021
Analyse the technology	22 February 2021	24 February 2021	11 March 2021
ITERATION 3: ***SAMPLE QUERIES 7, 8, 9***			
Run the queries for third iteration	25 February 2021	1 March 2021	13 March 2021
Analyse the queries for third iteration	2 March 2021	9 March 2021	19 March 2021
Analyse the technology	10 March 2021	17 March 2021	21 March 2021
Editing the dissertation document	29 March 2021	8 April 2021	5 April 2021
Deliverable 2 Deadline	22 April 2021	22 April 2021	22 April 2021

Professional, Legal, Ethical and Social Issues

Professional

All of the work which has been produced during this project has been carried out to a professional standard. In order to ensure that enough progress was being made throughout the project, weekly meetings with the supervisor were held to discuss the current work produced, and the work which still needed to be completed. It was important that no work was lost due to carelessness, such as overwriting an important file, and therefore the Google Drive cloud service was used to store all related project files. As an extra precaution, the entire folder was backed up to an external SSD hard drive on a regular basis to make sure that no data is lost.

Legal

As this project was a technology exploration project which involved analysing data, running queries on the data, and discussing the results of these queries, no personal data was collected during the project. The dataset being used was the Ergast Developer API, which states that the Ergast API can be used 'for personal, non-commercial applications and services including educational and research purposes' [3]. Also, the data visualisation tool Neo4j Bloom which was to be used during the project can be accessed via Neo4j Desktop for free for local databases instances with no download or activation being required. All other external sources have been checked that their use was suitable for this project, with references supplied where it is required.

[3] https://ergast.com/mrd/terms/

Ethical

This project has been ethically approved by Heriot Watt University to allow the project to be carried out. As a technology exploration project, it did not involve the use of any human subjects at any point and did not collect any sensitive or confidential information at any stage of the project.

Social

This project is aimed towards those who are using, or more importantly those who are considering the possibility of using, graph-based data modelling for their applications and services. This project should allow them to consider changing their current technologies if their use case suits graph-based data modelling.

Conclusion

The Neo4j tool is an extremely powerful tool which can be used to analyse large datasets in an effective and efficient way. Although the insertion of the data can be relatively slow at times, as even a relatively small csv file of half a million lines took multiple days of a Neo4 server running consecutively to be loaded in fully, it mostly makes up for this by being extremely quick at solving queries to find out various answers about a particular dataset. Furthermore, it offers a very useful method at graphically visualising the data, which usually makes it much easier to know precisely how a dataset is constructed and how it has been constructed. From initial uses of the Neo4j Bloom, albeit unfortunately not with the dataset explored in this project, it looks to be a very powerful plugin for Neo4j which offers an interesting way of displaying data in a format which is easy for many different users to understand what is being represented. Neo4j looks to be an extremely useful tool which can be used for analysing a wide range of datasets and looks to offer many advantages over traditional relational databases, including being able to visualise the data and having quick query response times.

There is also further scope where this project can be taken in multiple ways. Clearly, the world of Big Data is a very exciting prospect and new technologies to be able to analyse this ever-expanding world are being released at an increasing rate. This project has provided an insight into how effective a graph database tool, Neo4j, can be in analysing a specific dataset, and it is clear to see that graph database technologies are the future of Big Data. Further research into these graph database technologies, Neo4j and its alternatives, will show how beneficial they can be in the data analysis world and how their use should be implemented to allow us to explore data further.

Main Achievements

This project has proved to be successful in looking at how effective the Neo4j software is at analysing a particular dataset and whether it could be used in the future for other datasets. It has managed to achieve a number of things, such as:

- Looking at the effectiveness of how Neo4j loads in csv data, using multiple different files of various lengths
- Running a variety of different queries, of different types, to look at how well Neo4j can analyse queries about a particular dataset
- Looking at database visualisations of the dataset and see how it can make data easier to understand for a large range of different users
- In general, looking at the effectiveness of the Neo4j graph database technology to analyse a dataset, and whether its use could be beneficial in the future of data analysis

Limitations of the Work

It was not completely possible to view the data visualisation of the Formula One dataset using the Neo4j Bloom plugin. This is because it was not possible to use a local machine, and therefore the Neo4 Desktop application, as it quite simply was not powerful enough to be able to load in the length and size of csv files which were required for this project. This meant the project had to be carried out on the university server which, although it had its benefits such as being able to be run overnight without needing to leave the computer on, it was not the ideal solution as it did not allow for various plugins to be installed and analysed and the configuration could be not be changed as much as it could have been (as the server administrator needed to do this – although some settings were changed when requested). Although data visualisation was possible with the default Neo4j installation and some aspects could be analysed, the use of the Neo4j Bloom would have been very beneficial and would

have allowed for much more analysis of the Neo4j Bloom and the ability to visualise the data can be seen as a clear advantage of using Neo4j over traditional relational databases, which this study was unable to look at fully.

Possible Extensions and Future Work

Although this project has been successful at analysing the use of Neo4j with a specific F1 dataset, a number of different developments could be made.

In the future, it could be interesting to take a number of different datasets, of which contain a wide variety of different information which could be analysed. There could also be a number of interesting discussions which could lead from using many different csv files, and potentially other file types, such as using files with a wide variety of different lengths. How quickly can Neo4j load in a 5000-row file? What about 25000? 100000? Or even a 10 million row file? Having a number of different files with various lengths could provide a useful comparison of how effective Neo4j is at loading in certain files.

Another interesting development could be to create even more advanced and complex queries for a number of different datasets to look at the execution time of each of these queries and how effective Neo4j can be at analysing particular types of queries on a much wider range of datasets.

The study could be developed further by looking a number of different database technologies, such as SQL and traditional relational databases, and comparing the effectiveness of these technologies to load in data and analyse different types of queries compared to the Neo4j software. It could be intriguing to see whether relational databases are still considered better for some types of analysis tasks, or if graph databases have overtaken them completely.

References

Amazon Web Services, Inc. (2017). *What Is a Graph Database?* [Electronic Resource] Available at: https://aws.amazon.com/nosql/graph/.

Batty, M., 2013. Big data, smart cities and city planning. *Dialogues in human geography*, *3*(3), pp.274-279.

Ergast Developer API. *Ergast Developer API*. [Electronic Resource] Available at: https://ergast.com/mrd/ [Accessed 23 Sep. 2020].

Fernandes, D. and Bernardino, J., 2018, July. Graph Databases Comparison: AllegroGraph, ArangoDB, InfiniteGraph, Neo4J, and OrientDB. In *DATA* (pp. 373-380).

Grupo MAPFRE Corporativo - Acerca de MAPFRE. (2020). *Data analysis in Formula 1: the difference between victory and defeat*. [Electronic Resource] Available at: https://www.mapfre.com/en/data-analysis-in-formula-1-the-difference-between-victory-and-defeat/ [Accessed 7 Oct. 2020].

Guia, J., Soares, V.G. and Bernardino, J., 2017, January. Graph Databases: Neo4j Analysis. In *ICEIS (1)* (pp. 351-356).

Have, C.T. and Jensen, L.J., 2013. Are graph databases ready for bioinformatics? *Bioinformatics*, *29*(24), p.3107.

Holzschuher, F. and Peinl, R., 2013, March. Performance of graph query languages: comparison of cypher, gremlin and native access in Neo4j. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops* (pp. 195-204).

Jatana, N., Puri, S., Ahuja, M., Kathuria, I. and Gosain, D., 2012. A survey and comparison of relational and non-relational database. *International Journal of Engineering Research & Technology*, *1*(6), pp.1-5.

Nayak, A., Poriya, A. and Poojary, D., 2013. Type of NOSQL databases and its comparison with relational databases. *International Journal of Applied Information Systems*, *5*(4), pp.16-19.

Neo4j Graph Database Platform. (n.d.). *Neo4j Bloom User Interface Guide*. [Electronic Resource] Available at: https://neo4j.com/developer/neo4j-bloom/ [Accessed 25 Nov. 2020].

Neo4j Graph Database Platform. (2018). *Graph Databases for Beginners: Why Graph Technology Is the Future*. [Electronic Resource] Available at: https://neo4j.com/blog/why-graph-databases-are-the-future/.

Neo4j Graph Database Platform. (2019). *What Is a Graph Database and Property Graph | Neo4j*. [Electronic Resource] Available at: https://neo4j.com/developer/graph-database/.

Sharma, M., Sharma, V.D. and Bundele, M.M., 2018, November. Performance analysis of rdbms and no sql databases: Postgresql, mongodb and neo4j. In 2018 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE) (pp. 1-5). IEEE.

Sahatqija, K., Ajdari, J., Zenuni, X., Raufi, B. and Ismaili, F., 2018, May. Comparison between relational and NOSQL databases. In *2018 41st international convention on information and communication technology, electronics and microelectronics (MIPRO)* (pp. 0216-0221). IEEE.

Spenke, M. and Beilken, C. *InfoZoom - Analysing Formula One racing results with an interactive data mining and visualisation tool ISBN 1-85312-821-X*. [Electronic Resource] Available at: https://www.witpress.com/Secure/elibrary/papers/DATA00/DATA00044FU.pdf.

Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y. and Wilkins, D. (2010). A Comparison of a Graph Database and a Relational Database: A data provenance perspective. [Electronic Resource] Available at: https://www.researchgate.net/publication/220996559_A_comparison_of_a_graph_database_and_a_relational_database_A_data_provenance_perspective [Accessed 13 Apr. 2021].

Williamson, M. (2015). *A brief history of Formula One*. [Electronic Resource]. Available at: http://en.espn.co.uk/f1/motorsport/story/3831.html.

Wooden, A. *How Big Data And Analytics Power Formula 1*. [Electronic Resource] Intel. Available at: https://www.intel.co.uk/content/www/uk/en/it-management/cloud-analytic-hub/big-data-powers-f1.html.

www.tutorialspoint.com. (n.d.). *Google Colab - What is Google Colab? - Tutorialspoint*. [online] Available at: https://www.tutorialspoint.com/google_colab/what_is_google_colab.htm

Appendix A: Dataset Description

circuits.csv

Name of Attribute	*Variable Type*	*Description*
circuitId	Number - Integer	Unique identifier
circuitRef	String	Identifies the circuit
name	String	The name of the circuit
location	String	The city where the circuit is located
country	String	The country where the circuit is located
lat	Number – Float	The latitude of the circuit
lng	Number - Float	The longitude of the circuit
alt	Number – Integer	The altitude of the circuit
url	String	The URL where more information about the circuit can be found

constructors.csv

Name of Attribute	*Variable Type*	*Description*
constructorId	Number – Integer	Unique identifier
constructorRef	String	Identifies the constructor
name	String	The name of the constructor
nationality	String	The nationality of the constructor
url	String	The URL where more information about the constructor can be found

constructor_results.csv

Name of Attribute	*Variable Type*	*Description*
constructorResultsId	Number – Integer	Unique identifier
raceId	Number – Integer	The id of the race that occurred – linked to races.csv
constructorId	Number – Integer	The id of the constructor who took part in the race – linked to constructors.csv
points	Number – Integer	The number of points gained by the constructor in the race
status	String	The status of the constructor in the race

constructor_standings.csv

Name of Attribute	*Variable Type*	*Description*
constructorStandingsId	Number – Integer	Unique identifier
raceId	Number – Integer	The id of the race that occurred – linked to races.csv
constructorId	Number – Integer	The id of the constructor who took part in the race – linked to constructors.csv
points	Number – Integer	The number of points gained by the constructor in the season so far
position	Number – Integer	The current position of the constructor in the standings
positionText	String	The current position of the constructor in the standings, in text format
wins	Number – Integer	The number of wins by the constructor

drivers.csv

Name of Attribute	*Variable Type*	*Description*
driverId	Number – Integer	Unique identifier
driverRef	String	Identifies the driver
number	Number – Integer	The car number of the driver Note: permanent driver numbers are from 2014 season onwards
code	String	The three-letter code of the driver
forename	String	The first name of the driver
surname	String	The last name of the driver
dob	String	The date of birth of the driver
nationality	String	The nationality of the driver
url	String	The URL where more information about the driver can be found

races.csv

Name of Attribute	*Variable Type*	*Description*
raceId	Number – Integer	Unique identifier
year	Number – Integer	The year that the race took place
round	Number - Integer	The round that the race took place
circuitId	Number – Integer	The id of the circuit where the race took place – linked to circuits.csv
name	String	The official name of the race
date	String	The date the race took place
time	String	The start time of the race
url	String	The URL where more information about the race can be found

lap_times.csv

Note: lap time data is available from the 1996 season onwards.

Name of Attribute	*Variable Type*	*Description*
raceId	Number – Integer	The id of the race where the lap occurred – linked to races.csv
driverId	Number - Integer	The id of the driver who completed the lap – linked to drivers.csv
lap	Number – Integer	The number of the lap the driver completed
position	Number – Integer	The race position of the driver at the end of the lap
time	String	The lap time of the driver for the completed lap
milliseconds	Number – Integer	The lap time of the driver for the completed lap, in milliseconds

pit_stops.csv

Note: pit stop data is available from the 2012 season onwards.

Name of Attribute	*Variable Type*	*Description*
raceId	Number – Integer	The id of the race where the pitstop occurred – linked to races.csv
driverId	Number – Integer	The id of the driver who was involved in the pitstop – linked to drivers.csv
stop	Number – Integer	The number of the pitstop which the driver did in the race (i.e. 1st or 2nd pitstop)
lap	Number – Integer	The race lap which the driver stopped on to complete the pitstop
time	String	The time where the pitstop occurred
duration	Number – Float	The amount of time the driver spent in the pitlane during the pitstop
milliseconds	Number – Integer	The amount of time the driver spent in the pitlane during the pitstop, in milliseconds

qualifying.csv

Note: qualifying results are only fully supported from the 2003 season onwards.

Name of Attribute	*Variable Type*	*Description*
qualifyId	Number – Integer	Unique identifier
raceId	Number – Integer	The id of the race where the qualifying occurred – linked to races.csv
driverId	Number – Integer	The id of the driver who took part in the qualifying – linked to drivers.csv
constructorId	Number – Integer	The id of the constructor who took part in the qualifying – linked to constructors.csv
number	Number – Integer	The number of the car who took part in the qualifying
position	Number - Integer	The position which the driver finished in qualifying (note: the qualifying position may be different to the race start position, due to penalties / mechanical issues)
q1	String	The time set by the driver in the first qualifying session (all drivers)
q2	String	The time set by the driver in the second qualifying session (approximately the top 16 drivers from previous session)
q3	String	The time set by the driver in the third qualifying session (approximately the top 10 drivers from previous session)

seasons.csv

Name of Attribute	*Variable Type*	*Description*
year	Number – Integer	The year that the season took place
url	String	The URL where more information about the season can be found

status.csv

Name of Attribute	*Variable Type*	*Description*
statusId	Number – Integer	Unique identifier
status	String	The finishing state of the driver / car (e.g. disqualified or brake failure)

driver_standings.csv

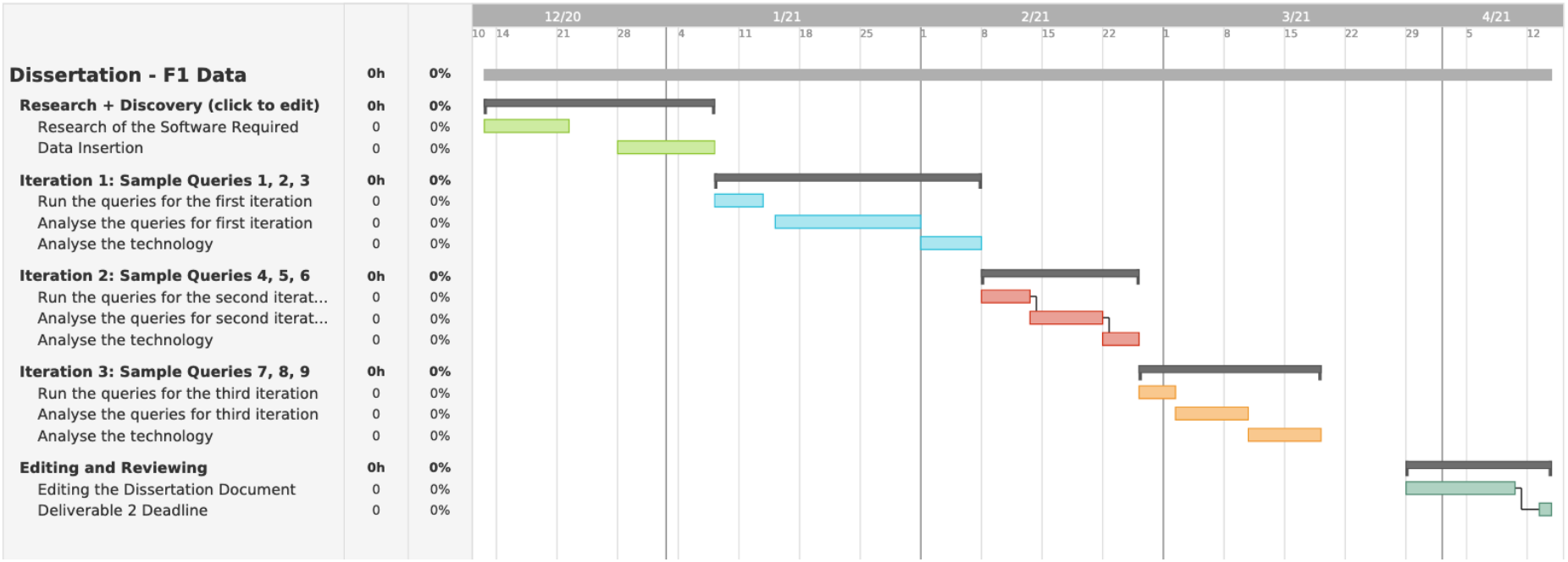
Name of Attribute	*Variable Type*	*Description*
driverStandingId	Number – Integer	Unique identifier
raceId	Number – Integer	The id of the race which took place – links to races.csv
driverId	Number – Integer	The id of the driver who took part in the race – links to drivers.csv
points	Number – Integer	The number of points gained by the driver in the season so far
position	Number – Integer	The position of the driver in the standings so far
positionText	String	The position of the driver in the standings so far, in text format Possible values: [integer (position), R (retired), D (disqualified), E (excluded), W (withdrawn), N (not classified)] Note: a grid position of 0 indicates starting from the pitlane
wins	Number – Integer	The number of wins by the driver

results.csv

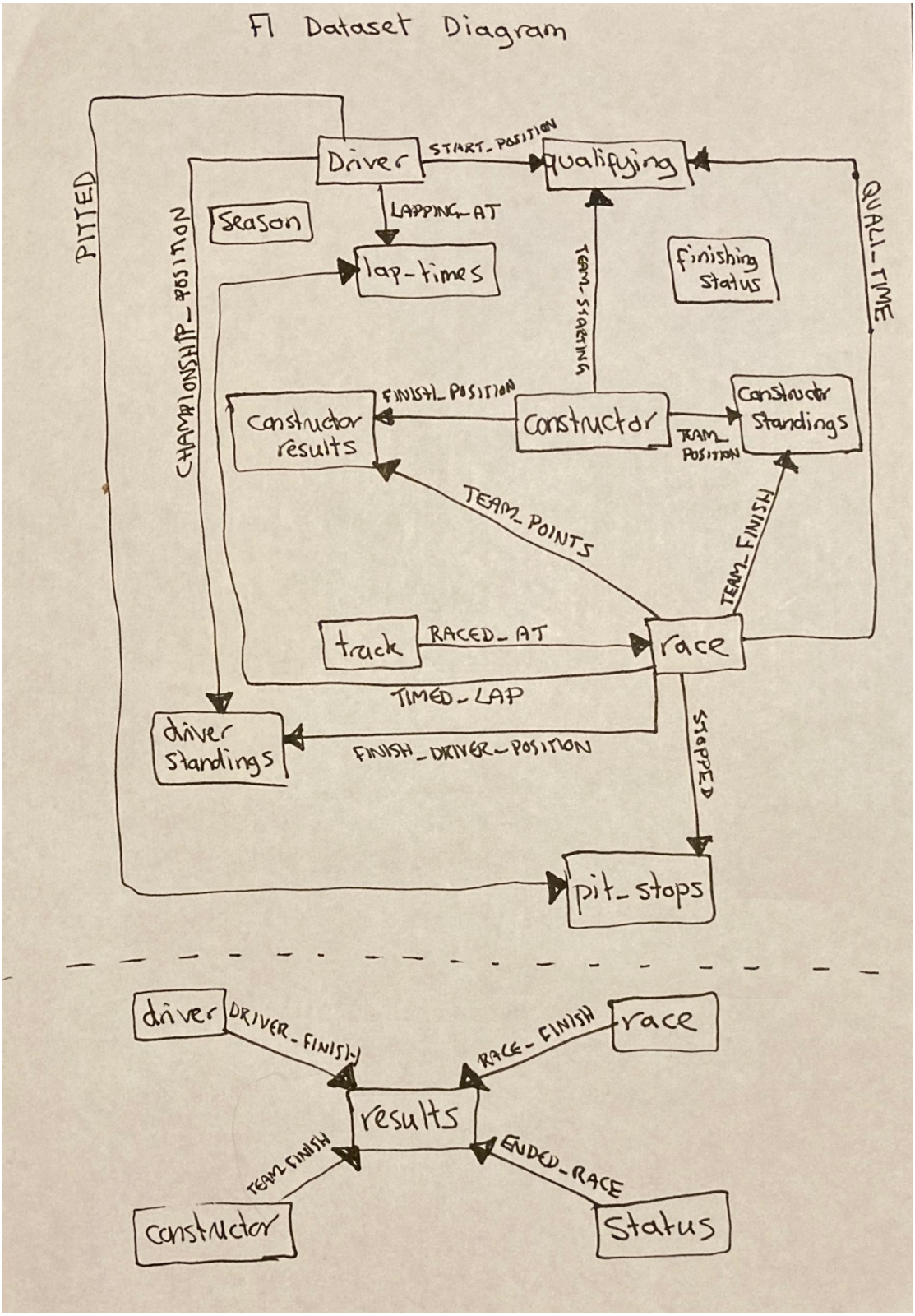
Name of Attribute	*Variable Type*	*Description*
resultId	Number – Integer	Unique identifier
raceId	Number – Integer	The id of the race which took place – linked to races.csv
driverId	Number – Integer	The id of the driver who took part in the race – linked to drivers.csv
constructorId	Number – Integer	The id of the constructor who took part in the race – linked to constructors.csv
number	Number – Integer	The number of the car who took part in the race
grid	Number – Integer	The starting position of the car who took part in the race
position	Number – Integer	The finish position of the car who took part in the race
positionText	String	The finish position of the car who took part in the race, in text format Possible values: [integer (position), R (retired), D (disqualified), E (excluded), W (withdrawn), N (not classified)]

		Note: a grid position of 0 indicates starting from the pitlane
positionOrder	Number – Integer	The finish position of the car who took part in the race, ignoring retirements / disqualifications
points	Number – Integer	The number of points gained by the driver in the race
laps	Number – Integer	The number of laps completed by the driver in the race
time	String	The total time taken by the driver to complete the race (note drivers in 2nd or below has the gap between them and the race winner)
milliseconds	Number – Integer	The total race time of the driver – in milliseconds)
fastestLap	Number - Integer	The lap number where the driver completed their fastest lap of the race Note: fastest laps times are included from the 2004 season onwards
rank	Number – Integer	The position of the driver of the fastest lap compared to the rest of the drivers in the race
fastestLapTime	String	The time of the fastest lap of the driver in the race
fastestLapSpeed	Number – Float	The speed of the driver during the fastest lap in the speed trap
statusId	Number – Integer	The finishing status of the driver in the race – linked to status.csv

Appendix B: Project Gantt Chart



Appendix C: F1 Dataset Diagram



Appendix D: Sample Queries

QUERY ONE

What is the optimal pitstop strategy of a race?

EXAMPLE(S)

e.g., which lap(s) did the race winner complete their pitstop on? How does this compare to the driver who finished last in the race?

DATASET FIELDS BEING USED

This looks at the driver's lap that they pitted vs their race position.

NOTES

Should you pit earlier on in the race (to be on fresher tyres than everyone else at the start) or leave it as late as possible to gain an advantage at the end.

QUERY TWO

What is the ideal number of pitstops for a driver during a race?

EXAMPLE(S)

e.g., does more pitstops during a race (and therefore fresher tyres for longer) lead to a better race position?

DATASET FIELDS BEING USED

This looks at the number of pitstops in a race by a driver vs the driver's race position.

NOTES

Could also look at the number of pitstops by a driver throughout the entire season, and how it compares to their final championship position.

QUERY THREE

Are quicker pitstops more likely to lead to higher race positions? Or is consistency more important?

EXAMPLE(S)

e.g., did the fastest pitstop of a race belong to the race winner?

e.g., what position did the driver with the slowest pitstop of the race finish?

DATASET FIELDS BEING USED

This looks at the driver's race position vs the pitstop time.

QUERY FOUR

Which country has the most championship points?

EXAMPLE(S)

e.g., for the 2005 season, which country would have won if all of the drivers of each country had their points total added together?

DATASET FIELDS BEING USED

This looks at the driver's nationality vs the championship points total of all the drivers.

NOTES

For a season where Michael Schumacher (German) won the Championship, would Germany have won the countries' world Championship?

QUERY FIVE

Which driver would win an overall World Championship since 1950?

EXAMPLE(S)

e.g., does Lewis Hamilton have the greatest number of championship totals in the history of formula one?

DATASET FIELDS BEING USED

This looks at the driver's championship points totals.

NOTES

The points system has changed numerous times in formula one since 1950. Would need to create one single points system and change all drivers points total to the new system.

QUERY SIX

Do drivers qualifying on pole position usually win the race?

EXAMPLE(S)

e.g., for the 2008 Brazilian GP, where did the driver on pole position finish the race?

e.g., for the entire 2008 season, how many drivers who qualified on pole position actually won the race? How many of them resulted in DNF (did not finish) classifications?

DATASET FIELDS BEING USED

This looks at the driver's qualifying position vs their race position / race status.

QUERY SEVEN

Do drivers usually win Grand Prix from their home country?

EXAMPLE(S)

e.g., do British drivers usually win the British GP?

e.g., do Spanish drivers usually win the Spanish GP?

DATASET FIELDS BEING USED

This looks at the driver's nationality vs the country that the circuit is located in.

NOTES

Could also look at the constructors' nationality instead of the drivers (i.e., do British constructors usually win the British GP?).

QUERY EIGHT

What lap is the fastest lap of a race usually set?

EXAMPLE(S)

e.g., was the fastest lap of the 2014 Australian GP set before lap 20?

e.g. was the fastest lap of the 2019 USA GP set after lap 50?

DATASET FIELDS BEING USED

This looks at the fastest lap of the driver.

NOTES

Is it better to set a fastest lap earlier on (to be on fresher tyres with more grip), or later in the race where you will have less fuel which you need to carry?

QUERY NINE

Do younger drivers win more races than older drivers?

EXAMPLE(S)

e.g., how many races have drivers under the age of 22 won?

e.g., how many championships have drivers over 35 won?

DATASET FIELDS BEING USED

This looks at the driver's age vs the number of race wins.

Appendix E: Textual Dataset Description

The first column of all tables in the dataset is a unique identifier for each specific row (record) in the table.

circuits.csv

This table covers all of the circuits which have been raced at throughout the history of Formula One, since 1950. It contains information of the name of the circuit, the city and country it is located in, the precise latitude and longitude, and URL links to find some more information.

Data Examples:

1, albert_park, Albert Park Grand Prix Circuit, Melbourne, Australia, -36.8497, 144.968, 10, http://en.wikipedia.org/wiki/Melbourne_Grand_Prix_Circuit

14, monza. Autodromo. Nazionale Di Monza, Monza, Italy, 45.6156, 9.28111, \N, http://en.wikipedia.org/wiki/Autodromo_Nazionale_Monza

constructors.csv

This part of the dataset covers all of the constructors (teams) who have taken part in Formula One at some point in history. It contains the name of the constructor, the nationality of where they originated, and URL links containing more information.

Data Examples:

9, red_bull, Red Bull, Austrian, http://en.wikipedia.org/wiki/Red_Bull_Racing

23, brawn, Brawn, British, http://en.wikipedia.org/wiki/Brawn_GP

105, maserati, Maserati, Italian, http://en.wikipedia.org/wiki/Maserati

constructor_results.csv

This piece of data links the race which took place and the constructor who took part and displays the number of points gained by the constructor, and the status of them in the race.

Data Examples:

Unique identifier, race identification, constructor identification, points, status

11, 18, 11, 0, \N

413, 56, 13, 0, \N

2182, 222, 24, 0, \N

constructor_standings.csv

This part of the dataset collects all of the positions, and number of points, of all of the different constructors who have taken part in a race. It contains the unique identifiers of the race which took place and the constructor who took part, to link them together, as well as the number of points they gained and what championship position they are currently in. The final column indicates the number of wins.

Data Examples:

unique identifier, race identification, constructor identification, points, position, position in text format, wins

3, 18, 3, 9, 2, 2, 0

45, 22, 6, 63, 1, 1, 4

25783, 40, 11, 0, 9, 9, 0

drivers.csv

This data contains information on all of the drivers who have taken part in Formula One, including their full name, unique driver number (only drivers from the 2014 season onwards have chosen their own number), date of birth, nationality and a URL to find further information.

Data Examples:

17, webber, \N, WEB, Mark, Webber, 27/08/1976, Australian, http://en.wikipedia.org/wiki/Mark_Webber

154, grosjean, 8, GRO, Romain, Grosjean, 17/04/1986, French, http://en.wikipedia.org/wiki/Romain_Grosjean

races.csv

This section of the dataset categorises all of the races which have taken place (unlike circuits.csv which just shows the actual tracks). It shows the year of the race, which circuit the race took place on, the date and time it took place, and a URL where more information can be found about the race events.

Data Examples:

Unique identification, year, round number, circuit identification, name, date, time, URL

20, 2008, 3, 3, Bahrain Grand Prix, 06/04/2008, 11:30:00,

http://en.wikipedia.org/wiki/2008_Bahrain_Grand_Prix

430, 1985, 11, 39, Dutch Grand Prix, 25/08/1985, \N,

http://en.wikipedia.org/wiki/1985_Dutch_Grand_Prix

lap_times.csv

This comprehensive part of the dataset displays every lap completed by each driver at every race they took part in. It links the race and driver together using their unique identifiers, and shows the lap number, current position they were in, and the lap time.

Data Examples:

Race identification, driver identification, lap number, driver position, lap time, lap time in milliseconds

842, 13, 16, 8, 01:44:1, 104098

964, 817, 1, 5, 01:41:7, 101724

pit_stops.csv

This section of the dataset shows all of the pitstops which have taken place throughout all of the races. It uses the unique identifiers of the driver and race to link them together, the stop number of the driver (e.g. their first stop of the race), the lap number of the pitstop, the time that the stop took place and how long the driver was in the pitlane for.

Data Examples:

Race identification, driver identification, stop number, lap number, time of stop, length of stop, length of stop in milliseconds

848, 15, 2, 34, 15:05:36, 22.131, 22131

991, 817, 2, 31, 15:05:15, 22.691, 22691

qualifying.csv

This piece of the dataset displays all of the qualification results of all race events. It uses the unique identifiers of the race, driver and constructor to link them together correctly, and shows the number, qualifying position and the final times from all 3 qualifying sessions.

Data Examples:

Unique identification, race identification, driver identification, constructor identification, number, race starting position, 1st session time, 2nd session time, 3rd session time

19, 13, 6, 2, 1, 01:35.3, 01:34.4, 01:35.7

5715, 893, 821, 15, 12, 10, 01:45.5, 01:44.2, \N

seasons.csv

This dataset simply classifies each season which has taken place. It contains the year and a URL for further information about the season.

Data Examples:

1978, https://en.wikipedia.org/wiki/1978_Formula_One_season

1955, https://en.wikipedia.org/wiki/1955_Formula_One_season

2015, https://en.wikipedia.org/wiki/2015_Formula_One_season

status.csv

This data collates all of the different finishing statuses of the drivers, for example if they have been disqualified or had a brakes failure which caused them to not finish the race.

Data Examples:

3, Accident

27, Tyre

60, Out of fuel

driver_standings.csv

This section of data gathers all of the points gained by drivers at each race event. It links the race and driver by their unique identifiers, and shows the number of points they have, their position, and number of wins.

Data Examples:

Unique identification, race identification, driver identification, number of points, position, position in text format, number of wins

23, 19, 21, 0, 15, 15, 0

245, 30, 1, 76, 1, 1, 4

results.csv

This dataset collects all of the data from a race event and collates it into this 1 set. It contains unique identifiers for the race, driver and constructor to link them together, and shows the number, grid position of the driver, the number of points gained, the number of laps the driver completed, the total race time taken, when they completed their fastest lap of the race, their fastest lap time and speed, and their finishing status.

Data Examples:

Unique identification, race identification, driver identification, constructor identification, number, grid position, finish position, finish position in text format, finish position without specifying if they finished or not, the number of points gained, number of laps completed, total time taken to finish, total time in milliseconds, what lap was their fastest, ranking based on fastest laps, their fastest lap time, maximum speed during their fastest lap, and their finishing status,

1, 18, 1, 1, 22, 1, 1, 1, 1, 10, 58, 34:50.6, 5690616, 39, 2, 01:27.5, 218.3, 1

66, 20, 20, 5, 15, 19, \N, R, 22, 0, 0, \N, \N, \N, \N, \N, \N, 5

24625, 1030, 840, 211, 18, 12, \N, R, 20, 0, 45, \N, \N, 25, 17, 01:43.3, 193.507, 23

Appendix F: Loading in the Data

```
// load in the drivers.csv file, changing the variable names - 848 nodes
LOAD CSV WITH HEADERS from 'file:///drivers.csv'
AS row
MERGE (d:Driver {driverId: row.number, forename: row.forename, surname: row.surname,
code: row.code, nationality: row.nationality, driverRef: row.driverRef, driverId: row.driverId,
dob: row.dob, url: row.url})
RETURN d;

// load in the constructors.csv file, changing the variable names - 210 nodes
LOAD CSV WITH HEADERS from 'file:///constructors.csv'
AS row WITH row WHERE row IS NOT NULL
MERGE (c:Constructor {constructorId: row.constructorId, constructorRef:
row.constructorRef, constructor: row.name, nationality: row.nationality, url: row.url})
RETURN c;

// load in the seasons.csv, changing the variable names - 71 nodes
LOAD CSV WITH HEADERS from 'file:///seasons.csv'
AS row
MERGE (s:Season {year: row.year, url: row.url})
RETURN s;

// load in the status.csv, changing the variable names - 135 nodes
LOAD CSV WITH HEADERS from 'file:///status.csv'
AS row
MERGE (f: finishingStatus {statusId: row.statusId, status: row. status})
RETURN f;

// load in the circuits.csv, changing the variable names - 74 nodes
LOAD CSV WITH HEADERS from 'file:///circuits.csv'
AS row
MERGE (t: track {circuitId: row.circuitId, circuitRef: row. circuitRef, name: row.name,
location: row.location, country: row.country, latitude: row.lat, longitude: row.lng, alt:
row.alt, url: row.url})
RETURN t;

// load in the races.csv, changing the variable names - 1031 nodes
LOAD CSV WITH HEADERS from 'file:///races.csv'
AS row
MERGE (r: race {raceId: row.raceId, year: row.year, round: row.round, circuitId:
row.circuitId, name: row.name, date: row.date, time: row.time, url: row.url})
RETURN r;

// add a relationship between circuits.csv and races.csv
```

```
// note: use MERGE instead of CREATE to avoid creating the same relationship twice if run multiple times
MATCH (t:track),(r:race)
WHERE t.circuitId = r.circuitId
MERGE (t)-[rel:RACED_AT]->(r)
RETURN type(rel)

// load in the constructor_results.csv, changing the variable names - 11,560 nodes
LOAD CSV WITH HEADERS from 'file:///constructor_results.csv'
AS row
MERGE (cr: constructorResults {constructorResultsId: row.constructorResultsId, raceId: row.raceId, constructorId: row.constructorId, points: row.points, status: row.status})
RETURN cr;

// add a relationship between constructor results, constructors and races - 23120 relationships
MATCH (cr:constructorResults),(c:Constructor),(r:race)
WHERE cr.raceId = r.raceId AND cr.constructorId = c.constructorId
MERGE (c)-[rel2:FINISH_POSITION]->(cr)
MERGE (r)-[rel3:TEAM_POINTS]->(cr)
RETURN type(rel2), type(rel3)

// load in the constructor_standings.csv, changing the variable names - 12326 nodes
LOAD CSV WITH HEADERS from 'file:///constructor_standings.csv'
AS row
MERGE (cs: constructorStandings {constructorStandingsId: row.constructorStandingsId, raceId: row.raceId, constructorId: row.constructorId, points: row.points, position: row.position, positionText: row.positionText, wins: row.wins})
RETURN cs;

// add a relationship between constructor standings, constructors and races - 24632 relationships
MATCH (cs:constructorStandings),(c:Constructor),(r:race)
WHERE cs.raceId = r.raceId AND cs.constructorId = c.constructorId
MERGE (c)-[rel4:TEAM_POSITION]->(cs)
MERGE (r)-[rel5:TEAM_FINISH]->(cs)
RETURN type(rel4), type(rel5)

// load in the pit_stops.csv, changing the variable names
LOAD CSV WITH HEADERS from 'file:///pit_stops.csv'
AS row
MERGE (pt: pitStops {raceId: row.raceId, driverId: row.driverId, stop: row.stop, lap: row.lap, time: row.time, duration: row.duration, milliseconds: row.milliseconds})
RETURN pt;

// add a relationship between pitstops, drivers and races
MATCH (pt:pitStops),(d:Driver),(r:race)
```

```
WHERE pt.raceId = r.raceId AND pt.driverId = d.driverId
MERGE (d)-[rel10:PITTED]->(pt)
MERGE (r)-[rel11:STOPPED]->(pt)
RETURN type(rel10), type(rel11)

// load in the qualifying.csv, changing the variable names
LOAD CSV WITH HEADERS from 'file:///qualifying.csv'
AS row
MERGE (qt: qualifying {qualifyId: row.qualifyId, raceId: row.raceId, driverId: row.driverId,
constructorId: row.constructorId, number: row.number, position: row.position, q1: row.q1,
q2: row.q2, q3: row.q3})
RETURN qt;

// add a relationship between qualifying, drivers, races and constructors
MATCH (qt:qualifying),(d:Driver),(r:race), (c:Constructor)
WHERE qt.raceId = r.raceId AND qt.driverId = d.driverId AND qt.constructorId =
c.constructorId
MERGE (d)-[rel12:START_POSITION]->(qt)
MERGE (r)-[rel13:QUALI_TIME]->(qt)
MERGE (c)-[rel14:TEAM_STARTING]->(qt)
RETURN type(rel12), type(rel13),  type(rel14)

// load in the results.csv, changing the variable names
LOAD CSV WITH HEADERS from 'file:///results.csv'
AS row
MERGE (rr: results {resultsId: row.resultsId, raceId: row.raceId, driverId: row.driverId,
constructorId: row.constructorId, number: row.number, grid: row.grid, position:
row.position, positionText: row.positionText, positionOrder: row.positionOrder, points:
row.points, laps: row.laps, time: row.time, milliseconds: row.milliseconds, fastestLap:
row.fastestLap, rank: row.rank, fastestLapTime: row.fastestLapTime, fastestLapSpeed:
row.fastestLapSpeed, statusId: row.statusId})
RETURN rr;

// add a relationship between results, drivers, races, constructors and status
MATCH (rr:results),(d:Driver),(r:race), (c:Constructor), (f:finishingStatus)
WHERE rr.raceId = r.raceId AND rr.driverId = d.driverId AND rr.constructorId =
c.constructorId AND rr.statusId = f.statusId
MERGE (d)-[rel15:DRIVER_FINISH]->(rr)
MERGE (r)-[rel16:RACE_FINISH]->(rr)
MERGE (c)-[rel17:TEAM_FINISH]->(rr)
MERGE (f)-[rel18:ENDED_RACE]->(rr)
RETURN type(rel15), type(rel16),  type(rel17), type(rel18)

// load in the driver_standings.csv, changing the variable names
LOAD CSV WITH HEADERS from 'file:///driver_standings.csv' AS row
```

```
MERGE (ds: driverStandings {driverStandingsId: row.driverStandingsId, raceId: row.raceId,
driverId: row.driverId, points: row.points, position: row.position, positionText:
row.positionText, wins: row.wins}) RETURN ds;

// add a relationship between driver standings, drivers and races
MATCH (ds:driverStandings),(d:Driver),(r:race) WHERE ds.raceId = r.raceId AND ds.driverId =
d.driverId MERGE (d)-[rel6:CHAMPIONSHIP_POSITION]->(ds) MERGE (r)-
[rel7:FINISH_DRIVER_POSITION]->(ds) RETURN type(rel6), type(rel7)

// load in the lap_times.csv, changing the variable names
PERIODIC COMMIT 25000
LOAD CSV WITH HEADERS from 'file:///lap_times.csv'
AS row
MERGE (lt: lapTimes {raceId: row.raceId, driverId: row.driverId, lap: row.lap, position:
row.position, time: row.time, milliseconds: row.milliseconds})
RETURN lt;

// add a relationship between laptimes, drivers and races
MATCH (lt:lapTimes),(d:Driver),(r:race)
WHERE lt.raceId = r.raceId AND lt.driverId = d.driverId
MERGE (d)-[rel8:LAPPING_AT]->(lt)
MERGE (r)-[rel9:TIMED_LAP]->(lt)
RETURN type(rel8), type(rel9)
```

Appendix G: Query to Return All British Drivers

```
%cypher MATCH (d:Driver) WHERE d.nationality = "British" RETURN
d.forename AS forename, d.surname AS surname, d.dob AS dateOfBirth
```

forename	surname	dateOfBirth
Lewis	Hamilton	1985-01-07
David	Coulthard	1971-03-27
Jenson	Button	1980-01-19
Anthony	Davidson	1979-04-18
Justin	Wilson	1978-07-31
Eddie	Irvine	1965-11-10
Johnny	Herbert	1964-06-25
Allan	McNish	1969-12-29
Damon	Hill	1960-09-17
Martin	Brundle	1959-06-01
Mark	Blundell	1966-04-08
Nigel	Mansell	1953-08-08
Derek	Warwick	1954-08-27
Perry	McCarthy	1961-03-03
Julian	Bailey	1961-10-09
Martin	Donnelly	1964-03-26
Jonathan	Palmer	1956-11-07
Johnny	Dumfries	1958-04-26
Kenny	Acheson	1957-11-27
John	Watson	1946-05-04
Brian	Henton	1946-09-19
Geoff	Lees	1951-05-01
Rupert	Keegan	1955-02-26
Stephen	South	1952-02-19
Tiff	Needell	1951-10-29
James	Hunt	1947-08-29
Divina	Galica	1944-08-13
Tony	Trimmer	1943-01-24
Tom	Pryce	1949-06-11
David	Purley	1945-01-26
Jackie	Oliver	1942-08-14
Andy	Sutcliffe	1947-05-09
Guy	Edwards	1942-12-30
Ian	Ashley	1947-10-26
Bob	Evans	1947-06-11

forename	surname	dateOfBirth
Damien	Magee	1945-11-17
Mike	Wilds	1946-01-07
Graham	Hill	1929-02-15
Tony	Brise	1952-03-28
Dave	Morgan	1944-08-07
Jim	Crawford	1948-02-13
Mike	Hailwood	1940-04-02
Richard	Robarts	1944-09-22
Brian	Redman	1937-03-09
Peter	Gethin	1940-02-21
Derek	Bell	1941-10-31
David	Hobbs	1939-06-09
Jackie	Stewart	1939-06-11
Mike	Beuttler	1940-04-13
Roger	Williamson	1948-02-02
John	Surtees	1934-02-11
Vic	Elford	1935-06-10
Chris	Craft	1939-11-17
John	Miles	1943-06-14
Piers	Courage	1942-05-27
Peter	Westbury	1938-05-26
Richard	Attwood	1940-04-04
Jim	Clark	1936-03-04
Mike	Spence	1936-12-30
Robin	Widdows	1942-05-27
Bob	Anderson	1931-05-19
Mike	Parkes	1931-09-24
Chris	Irwin	1942-06-27
Alan	Rees	1938-01-12
Brian	Hart	1936-09-07
Jonathan	Williams	1942-10-26
Peter	Arundell	1933-11-08
Vic	Wilson	1931-04-14
John	Taylor	1933-03-23
Chris	Lawrence	1933-07-27
Trevor	Taylor	1936-12-26
Innes	Ireland	1930-06-12
David	Prophet	1937-10-09
John	Rhodes	1927-08-18
Ian	Raby	1921-09-22

forename	surname	dateOfBirth
Alan	Rollinson	1943-05-15
Brian	Gubby	1934-04-17
John	Campbell-Jones	1930-01-21
Ian	Burgess	1930-07-06
Tim	Parnell	1932-06-25
Jackie	Lewis	1936-11-01
Roy	Salvadori	1922-05-12
Tony	Marsh	1931-07-20
Gerry	Ashmore	1936-07-25
Colin	Davis	1933-07-29
Keith	Greene	1938-01-05
Rob	Schroeder	1926-05-11
Stirling	Moss	1929-09-17
Cliff	Allison	1932-02-08
Tony	Brooks	1932-02-25
Henry	Taylor	1932-12-16
Brian	Naylor	1923-03-24
Jack	Fairman	1913-03-15
Geoff	Duke	1923-03-29
Ken	Miles	1918-11-01
Alan	Stacey	1933-08-29
Chris	Bristow	1937-12-02
Bruce	Halford	1931-05-18
Mike	Taylor	1934-04-24
Ron	Flockhart	1923-06-16
David	Piper	1930-12-02
Arthur	Owen	1915-03-23
Horace	Gould	1918-09-20
Ivor	Bueb	1923-06-06
Peter	Ashdown	1934-10-16
Bill	Moss	1933-09-04
Dennis	Taylor	1921-06-12
Mike	Hawthorn	1929-04-10
Peter	Collins	1931-11-06
Paul	Emery	1916-11-12
Bernie	Ecclestone	1930-10-28
Stuart	Lewis-Evans	1930-04-20
Dick	Gibson	1918-04-16
Tom	Bridger	1934-06-24
Les	Leston	1920-12-16

forename	surname	dateOfBirth
Mike	MacDowel	1932-09-13
Bob	Gerard	1914-01-19
Colin	Chapman	1928-05-19
Desmond	Titterington	1928-05-01
Archie	Scott Brown	1927-05-13
Lance	Macklin	1919-09-02
Ted	Whiteaway	1928-11-01
Peter	Walker	1912-10-07
Ken	Wharton	1916-03-21
Kenneth	McAlpine	1920-09-21
Leslie	Marr	1922-08-14
Tony	Rolt	1918-10-16
Don	Beauman	1928-07-26
Leslie	Thorne	1916-06-23
Bill	Whitehouse	1909-04-01
John	Riseley-Prichard	1924-01-17
Reg	Parnell	1911-07-02
Peter	Whitehead	1914-11-12
Eric	Brandon	1920-07-18
Alan	Brown	1919-11-20
Rodney	Nuckey	1929-06-26
John	Barber	1929-07-22
Tony	Crook	1920-02-16
Jimmy	Stewart	1931-03-06
Ian	Stewart	1929-07-15
Duncan	Hamilton	1920-04-30
George	Abecassis	1913-03-21
Robin	Montgomerie-Charrington	1915-06-23
Dennis	Poore	1916-08-19
Eric	Thompson	1919-11-04
Ken	Downing	1917-12-05
Graham	Whitehead	1922-04-15
David	Murray	1909-12-28
Bill	Aston	1900-03-29
Philip	Fotheringham-Parker	1907-09-22
Brian	Shawe Taylor	1915-01-28
John	James	1914-05-10
Ken	Richardson	1911-08-21
David	Hampshire	1917-12-29
Geoff	Crossley	1921-05-11

forename	surname	dateOfBirth
Cuth	Harrison	1906-07-06
Joe	Fry	1915-10-26
Leslie	Johnson	1912-03-22
Paul	di Resta	1986-04-16
Max	Chilton	1991-04-21
Will	Stevens	1991-06-28
Jolyon	Palmer	1991-01-20
Lando	Norris	1999-11-13
George	Russell	1998-02-15

Appendix H: Extract of Output of Query to Return Fastest Lap Data

```
-----------------------------------------------------------------------------------------------------------------+
| forename    | surname       | racePosition | fastestLap | fastestLapTime | raceName
| yearHeld |
+-----------------------------------------------------------------------------------------------------------------+
| "Lewis"     | "Hamilton"     | "1"          | "39"       | "1:27.452"     | "Australian Grand Prix"
| "2008"   |
| "Nick"      | "Heidfeld"     | "2"          | "41"       | "1:27.739"     | "Australian Grand Prix"    |
"2008"   |
| "Nico"      | "Rosberg"      | "3"          | "41"       | "1:28.090"     | "Australian Grand Prix"    |
"2008"   |
| "Fernando"   | "Alonso"       | "4"          | "58"       | "1:28.603"     | "Australian Grand Prix"
| "2008"   |
| "Heikki"    | "Kovalainen"    | "5"          | "43"       | "1:27.418"     | "Australian Grand Prix"
| "2008"   |
| "Kazuki"    | "Nakajima"     | "6"          | "50"       | "1:29.639"     | "Australian Grand Prix"
| "2008"   |
| "Sébastien"  | "Bourdais"     | "7"          | "22"       | "1:29.534"     | "Australian Grand Prix"
| "2008"   |
| "Kimi"      | "Räikkönen"    | "8"          | "20"       | "1:27.903"     | "Australian Grand Prix"
| "2008"   |
| "Robert"    | "Kubica"       | "\N"         | "15"       | "1:28.753"     | "Australian Grand Prix"
| "2008"   |
| "Timo"      | "Glock"        | "\N"         | "23"       | "1:29.558"     | "Australian Grand Prix"    |
"2008"   |
| "Takuma"    | "Sato"         | "\N"         | "24"       | "1:30.892"     | "Australian Grand Prix"
| "2008"   |
| "Nelson"    | "Piquet Jr."    | "\N"         | "20"       | "1:31.384"     | "Australian Grand Prix"
| "2008"   |
| "Felipe"    | "Massa"        | "\N"         | "23"       | "1:28.175"     | "Australian Grand Prix"    |
"2008"   |
| "David"     | "Coulthard"    | "\N"         | "21"       | "1:29.502"     | "Australian Grand Prix"
| "2008"   |
| "Jarno"     | "Trulli"       | "\N"         | "18"       | "1:29.310"     | "Australian Grand Prix"    |
"2008"   |
| "Adrian"    | "Sutil"        | "\N"         | "8"        | "1:32.021"     | "Australian Grand Prix"    |
"2008"   |
| "Mark"      | "Webber"       | "\N"         | "\N"       | "\N"           | "Australian Grand Prix"    |
"2008"   |
| "Jenson"    | "Button"       | "\N"         | "\N"       | "\N"           | "Australian Grand Prix"    |
"2008"   |
| "Anthony"   | "Davidson"     | "\N"         | "\N"       | "\N"           | "Australian Grand Prix"    |
"2008"   |
| "Sebastian"  | "Vettel"       | "\N"         | "\N"       | "\N"           | "Australian Grand Prix"    |
"2008"   |
```

"Giancarlo"	"Fisichella"	"\N"	"\N"	"\N"	"Australian Grand Prix"	"2008"
"Rubens"	"Barrichello"	"\N"	"44"	"1:28.736"	"Australian Grand Prix"	"2008"
"Kimi"	"Räikkönen"	"1"	"37"	"1:35.405"	"Malaysian Grand Prix"	"2008"
"Robert"	"Kubica"	"2"	"39"	"1:35.921"	"Malaysian Grand Prix"	"2008"
"Heikki"	"Kovalainen"	"3"	"19"	"1:35.922"	"Malaysian Grand Prix"	"2008"
"Jarno"	"Trulli"	"4"	"53"	"1:36.068"	"Malaysian Grand Prix"	"2008"
"Lewis"	"Hamilton"	"5"	"53"	"1:35.462"	"Malaysian Grand Prix"	"2008"
"Nick"	"Heidfeld"	"6"	"55"	"1:35.366"	"Malaysian Grand Prix"	"2008"
"Mark"	"Webber"	"7"	"53"	"1:36.696"	"Malaysian Grand Prix"	"2008"
"Fernando"	"Alonso"	"8"	"40"	"1:36.288"	"Malaysian Grand Prix"	"2008"
"David"	"Coulthard"	"9"	"55"	"1:36.206"	"Malaysian Grand Prix"	"2008"
"Jenson"	"Button"	"10"	"56"	"1:35.715"	"Malaysian Grand Prix"	"2008"
"Nelson"	"Piquet Jr."	"11"	"52"	"1:36.956"	"Malaysian Grand Prix"	"2008"
"Giancarlo"	"Fisichella"	"12"	"52"	"1:36.962"	"Malaysian Grand Prix"	"2008"
"Rubens"	"Barrichello"	"13"	"55"	"1:36.693"	"Malaysian Grand Prix"	"2008"
"Nico"	"Rosberg"	"14"	"55"	"1:36.782"	"Malaysian Grand Prix"	"2008"
"Anthony"	"Davidson"	"15"	"55"	"1:38.171"	"Malaysian Grand Prix"	"2008"
"Takuma"	"Sato"	"16"	"53"	"1:38.504"	"Malaysian Grand Prix"	"2008"
"Kazuki"	"Nakajima"	"17"	"19"	"1:37.711"	"Malaysian Grand Prix"	"2008"
"Sebastian"	"Vettel"	"\N"	"37"	"1:36.870"	"Malaysian Grand Prix"	"2008"
"Felipe"	"Massa"	"\N"	"15"	"1:35.914"	"Malaysian Grand Prix"	"2008"
"Adrian"	"Sutil"	"\N"	"3"	"1:40.330"	"Malaysian Grand Prix"	"2008"
"Timo"	"Glock"	"\N"	"\N"	"\N"	"Malaysian Grand Prix"	"2008"

"Sébastien"	"Bourdais"	"\N"	"\N"	"\N"	"Malaysian Grand Prix"	"2008"
"Felipe"	"Massa"	"1"	"38"	"1:33.600"	"Bahrain Grand Prix"	"2008"
"Kimi"	"Räikkönen"	"2"	"35"	"1:33.709"	"Bahrain Grand Prix"	"2008"
"Robert"	"Kubica"	"3"	"55"	"1:33.775"	"Bahrain Grand Prix"	"2008"
"Nick"	"Heidfeld"	"4"	"48"	"1:33.565"	"Bahrain Grand Prix"	"2008"
"Heikki"	"Kovalainen"	"5"	"49"	"1:33.193"	"Bahrain Grand Prix"	"2008"
"Jarno"	"Trulli"	"6"	"45"	"1:34.204"	"Bahrain Grand Prix"	"2008"
"Mark"	"Webber"	"7"	"51"	"1:34.305"	"Bahrain Grand Prix"	"2008"
"Nico"	"Rosberg"	"8"	"57"	"1:34.072"	"Bahrain Grand Prix"	"2008"
"Timo"	"Glock"	"9"	"56"	"1:34.807"	"Bahrain Grand Prix"	"2008"
"Fernando"	"Alonso"	"10"	"35"	"1:35.194"	"Bahrain Grand Prix"	"2008"
"Rubens"	"Barrichello"	"11"	"40"	"1:34.855"	"Bahrain Grand Prix"	"2008"
"Giancarlo"	"Fisichella"	"12"	"56"	"1:35.057"	"Bahrain Grand Prix"	"2008"
"Lewis"	"Hamilton"	"13"	"25"	"1:35.520"	"Bahrain Grand Prix"	"2008"
"Kazuki"	"Nakajima"	"14"	"30"	"1:35.433"	"Bahrain Grand Prix"	"2008"
"Sébastien"	"Bourdais"	"15"	"33"	"1:35.333"	"Bahrain Grand Prix"	"2008"
"Anthony"	"Davidson"	"16"	"45"	"1:35.324"	"Bahrain Grand Prix"	"2008"
"Takuma"	"Sato"	"17"	"56"	"1:35.891"	"Bahrain Grand Prix"	"2008"
"David"	"Coulthard"	"18"	"43"	"1:35.351"	"Bahrain Grand Prix"	"2008"
"Adrian"	"Sutil"	"19"	"55"	"1:35.442"	"Bahrain Grand Prix"	"2008"
"Nelson"	"Piquet Jr."	"\N"	"31"	"1:35.129"	"Bahrain Grand Prix"	"2008"
"Jenson"	"Button"	"\N"	"11"	"1:36.125"	"Bahrain Grand Prix"	"2008"
"Sebastian"	"Vettel"	"\N"	"\N"	"\N"	"Bahrain Grand Prix"	"2008"

"Kimi"	"Räikkönen"	"1"	"46"	"1:21.670"	"Spanish Grand Prix"	"2008"
"Felipe"	"Massa"	"2"	"45"	"1:21.801"	"Spanish Grand Prix"	"2008"
"Lewis"	"Hamilton"	"3"	"20"	"1:22.017"	"Spanish Grand Prix"	"2008"
"Robert"	"Kubica"	"4"	"20"	"1:22.106"	"Spanish Grand Prix"	"2008"
"Mark"	"Webber"	"5"	"19"	"1:22.564"	"Spanish Grand Prix"	"2008"
"Jenson"	"Button"	"6"	"66"	"1:22.353"	"Spanish Grand Prix"	"2008"
"Kazuki"	"Nakajima"	"7"	"48"	"1:23.549"	"Spanish Grand Prix"	"2008"
"Jarno"	"Trulli"	"8"	"45"	"1:22.758"	"Spanish Grand Prix"	"2008"
"Nick"	"Heidfeld"	"9"	"21"	"1:22.519"	"Spanish Grand Prix"	"2008"
"Giancarlo"	"Fisichella"	"10"	"40"	"1:23.439"	"Spanish Grand Prix"	"2008"
"Timo"	"Glock"	"11"	"57"	"1:23.007"	"Spanish Grand Prix"	"2008"
"David"	"Coulthard"	"12"	"57"	"1:22.842"	"Spanish Grand Prix"	"2008"
"Takuma"	"Sato"	"13"	"34"	"1:24.617"	"Spanish Grand Prix"	"2008"
"Nico"	"Rosberg"	"\N"	"20"	"1:23.319"	"Spanish Grand Prix"	"2008"
"Fernando"	"Alonso"	"\N"	"15"	"1:22.683"	"Spanish Grand Prix"	"2008"
"Rubens"	"Barrichello"	"\N"	"21"	"1:23.858"	"Spanish Grand Prix"	"2008"
"Heikki"	"Kovalainen"	"\N"	"19"	"1:22.453"	"Spanish Grand Prix"	"2008"
"Anthony"	"Davidson"	"\N"	"6"	"1:26.864"	"Spanish Grand Prix"	"2008"
"Sébastien"	"Bourdais"	"\N"	"6"	"1:25.999"	"Spanish Grand Prix"	"2008"
"Nelson"	"Piquet Jr."	"\N"	"6"	"1:25.444"	"Spanish Grand Prix"	"2008"
"Adrian"	"Sutil"	"\N"	"\N"	"\N"	"Spanish Grand Prix"	"2008"
"Sebastian"	"Vettel"	"\N"	"\N"	"\N"	"Spanish Grand Prix"	"2008"
"Felipe"	"Massa"	"1"	"16"	"1:26.666"	"Turkish Grand Prix"	"2008"

"Lewis"	"Hamilton"	"2"	"31"	"1:26.529"	"Turkish Grand Prix"	"2008"
"Kimi"	"Räikkönen"	"3"	"20"	"1:26.506"	"Turkish Grand Prix"	"2008"
"Robert"	"Kubica"	"4"	"17"	"1:26.780"	"Turkish Grand Prix"	"2008"
"Nick"	"Heidfeld"	"5"	"20"	"1:27.219"	"Turkish Grand Prix"	"2008"
"Fernando"	"Alonso"	"6"	"57"	"1:27.280"	"Turkish Grand Prix"	"2008"
"Mark"	"Webber"	"7"	"14"	"1:27.630"	"Turkish Grand Prix"	"2008"
"Nico"	"Rosberg"	"8"	"55"	"1:27.795"	"Turkish Grand Prix"	"2008"
"David"	"Coulthard"	"9"	"41"	"1:27.966"	"Turkish Grand Prix"	"2008"
"Jarno"	"Trulli"	"10"	"52"	"1:27.926"	"Turkish Grand Prix"	"2008"
"Jenson"	"Button"	"11"	"57"	"1:27.998"	"Turkish Grand Prix"	"2008"
"Heikki"	"Kovalainen"	"12"	"55"	"1:27.640"	"Turkish Grand Prix"	"2008"
"Timo"	"Glock"	"13"	"28"	"1:28.303"	"Turkish Grand Prix"	"2008"
"Rubens"	"Barrichello"	"14"	"56"	"1:28.017"	"Turkish Grand Prix"	"2008"
"Nelson"	"Piquet Jr."	"15"	"40"	"1:27.867"	"Turkish Grand Prix"	"2008"
"Adrian"	"Sutil"	"16"	"35"	"1:28.780"	"Turkish Grand Prix"	"2008"
"Sebastian"	"Vettel"	"17"	"38"	"1:28.180"	"Turkish Grand Prix"	"2008"
"Sébastien"	"Bourdais"	"\N"	"19"	"1:28.745"	"Turkish Grand Prix"	"2008"
"Kazuki"	"Nakajima"	"\N"	"\N"	"\N"	"Turkish Grand Prix"	"2008"
"Giancarlo"	"Fisichella"	"\N"	"\N"	"\N"	"Turkish Grand Prix"	"2008"
"Lewis"	"Hamilton"	"1"	"71"	"1:18.510"	"Monaco Grand Prix"	"2008"
"Robert"	"Kubica"	"2"	"75"	"1:17.933"	"Monaco Grand Prix"	"2008"
"Felipe"	"Massa"	"3"	"76"	"1:17.886"	"Monaco Grand Prix"	"2008"
"Mark"	"Webber"	"4"	"74"	"1:19.036"	"Monaco Grand Prix"	"2008"

"Sebastian"	"Vettel"	"5"	"74"	"1:18.787"	"Monaco Grand Prix"	"2008"
"Rubens"	"Barrichello"	"6"	"75"	"1:19.574"	"Monaco Grand Prix"	"2008"
"Kazuki"	"Nakajima"	"7"	"75"	"1:19.910"	"Monaco Grand Prix"	"2008"
"Heikki"	"Kovalainen"	"8"	"74"	"1:17.282"	"Monaco Grand Prix"	"2008"
"Kimi"	"Räikkönen"	"9"	"74"	"1:16.689"	"Monaco Grand Prix"	"2008"
"Fernando"	"Alonso"	"10"	"73"	"1:17.869"	"Monaco Grand Prix"	"2008"
"Jenson"	"Button"	"11"	"74"	"1:19.582"	"Monaco Grand Prix"	"2008"
"Timo"	"Glock"	"12"	"75"	"1:19.618"	"Monaco Grand Prix"	"2008"
"Jarno"	"Trulli"	"13"	"75"	"1:19.830"	"Monaco Grand Prix"	"2008"
"Nick"	"Heidfeld"	"14"	"72"	"1:20.251"	"Monaco Grand Prix"	"2008"
"Adrian"	"Sutil"	"\N"	"60"	"1:22.039"	"Monaco Grand Prix"	"2008"
"Nico"	"Rosberg"	"\N"	"58"	"1:21.270"	"Monaco Grand Prix"	"2008"
"Nelson"	"Piquet Jr."	"\N"	"40"	"1:31.187"	"Monaco Grand Prix"	"2008"
"Giancarlo"	"Fisichella"	"\N"	"33"	"1:32.849"	"Monaco Grand Prix"	"2008"
"David"	"Coulthard"	"\N"	"7"	"1:42.112"	"Monaco Grand Prix"	"2008"
"Sébastien"	"Bourdais"	"\N"	"7"	"1:41.150"	"Monaco Grand Prix"	"2008"
"Robert"	"Kubica"	"1"	"47"	"1:17.539"	"Canadian Grand Prix"	"2008"
"Nick"	"Heidfeld"	"2"	"25"	"1:17.430"	"Canadian Grand Prix"	"2008"
"David"	"Coulthard"	"3"	"36"	"1:18.085"	"Canadian Grand Prix"	"2008"
"Timo"	"Glock"	"4"	"31"	"1:19.087"	"Canadian Grand Prix"	"2008"
"Felipe"	"Massa"	"5"	"11"	"1:18.006"	"Canadian Grand Prix"	"2008"
"Jarno"	"Trulli"	"6"	"38"	"1:18.870"	"Canadian Grand Prix"	"2008"
"Rubens"	"Barrichello"	"7"	"35"	"1:18.301"	"Canadian Grand Prix"	"2008"

"Sebastian"	"Vettel"	"8"	"33"	"1:18.532"	"Canadian Grand Prix"	"2008"
"Heikki"	"Kovalainen"	"9"	"62"	"1:18.462"	"Canadian Grand Prix"	"2008"
"Nico"	"Rosberg"	"10"	"14"	"1:17.977"	"Canadian Grand Prix"	"2008"
"Jenson"	"Button"	"11"	"58"	"1:19.352"	"Canadian Grand Prix"	"2008"
"Mark"	"Webber"	"12"	"51"	"1:18.201"	"Canadian Grand Prix"	"2008"
"Sébastien"	"Bourdais"	"13"	"34"	"1:18.620"	"Canadian Grand Prix"	"2008"
"Giancarlo"	"Fisichella"	"\N"	"41"	"1:19.066"	"Canadian Grand Prix"	"2008"
"Kazuki"	"Nakajima"	"\N"	"30"	"1:18.784"	"Canadian Grand Prix"	"2008"
"Fernando"	"Alonso"	"\N"	"16"	"1:18.225"	"Canadian Grand Prix"	"2008"
"Nelson"	"Piquet Jr."	"\N"	"16"	"1:19.239"	"Canadian Grand Prix"	"2008"
"Kimi"	"Räikkönen"	"\N"	"14"	"1:17.387"	"Canadian Grand Prix"	"2008"
"Lewis"	"Hamilton"	"\N"	"4"	"1:17.506"	"Canadian Grand Prix"	"2008"
"Adrian"	"Sutil"	"\N"	"7"	"1:20.666"	"Canadian Grand Prix"	"2008"
"Felipe"	"Massa"	"1"	"20"	"1:16.729"	"French Grand Prix"	"2008"
"Kimi"	"Räikkönen"	"2"	"16"	"1:16.630"	"French Grand Prix"	"2008"
"Jarno"	"Trulli"	"3"	"12"	"1:17.567"	"French Grand Prix"	"2008"
"Heikki"	"Kovalainen"	"4"	"46"	"1:17.134"	"French Grand Prix"	"2008"
"Robert"	"Kubica"	"5"	"16"	"1:17.172"	"French Grand Prix"	"2008"
"Mark"	"Webber"	"6"	"22"	"1:17.507"	"French Grand Prix"	"2008"
"Nelson"	"Piquet Jr."	"7"	"24"	"1:17.758"	"French Grand Prix"	"2008"
"Fernando"	"Alonso"	"8"	"39"	"1:17.641"	"French Grand Prix"	"2008"
"David"	"Coulthard"	"9"	"23"	"1:17.818"	"French Grand Prix"	"2008"
"Lewis"	"Hamilton"	"10"	"40"	"1:17.453"	"French Grand Prix"	"2008"

"Timo"	"Glock"	"11"	"45"	"1:17.836"	"French Grand Prix"	"2008"
"Sebastian"	"Vettel"	"12"	"36"	"1:17.760"	"French Grand Prix"	"2008"
"Nick"	"Heidfeld"	"13"	"46"	"1:17.716"	"French Grand Prix"	"2008"
"Rubens"	"Barrichello"	"14"	"54"	"1:17.969"	"French Grand Prix"	"2008"
"Kazuki"	"Nakajima"	"15"	"60"	"1:18.054"	"French Grand Prix"	"2008"
"Nico"	"Rosberg"	"16"	"38"	"1:18.311"	"French Grand Prix"	"2008"
"Sébastien"	"Bourdais"	"17"	"28"	"1:18.216"	"French Grand Prix"	"2008"
"Giancarlo"	"Fisichella"	"18"	"53"	"1:18.557"	"French Grand Prix"	"2008"
"Adrian"	"Sutil"	"19"	"41"	"1:18.462"	"French Grand Prix"	"2008"
"Jenson"	"Button"	"\N"	"15"	"1:20.876"	"French Grand Prix"	"2008"
"Lewis"	"Hamilton"	"1"	"16"	"1:32.817"	"British Grand Prix"	"2008"
"Nick"	"Heidfeld"	"2"	"21"	"1:32.719"	"British Grand Prix"	"2008"
"Rubens"	"Barrichello"	"3"	"21"	"1:33.386"	"British Grand Prix"	"2008"
"Kimi"	"Räikkönen"	"4"	"18"	"1:32.150"	"British Grand Prix"	"2008"
"Heikki"	"Kovalainen"	"5"	"17"	"1:33.130"	"British Grand Prix"	"2008"
"Fernando"	"Alonso"	"6"	"17"	"1:33.133"	"British Grand Prix"	"2008"
"Jarno"	"Trulli"	"7"	"22"	"1:33.808"	"British Grand Prix"	"2008"
"Kazuki"	"Nakajima"	"8"	"21"	"1:34.277"	"British Grand Prix"	"2008"
"Nico"	"Rosberg"	"9"	"21"	"1:34.797"	"British Grand Prix"	"2008"
"Mark"	"Webber"	"10"	"17"	"1:32.952"	"British Grand Prix"	"2008"
"Sébastien"	"Bourdais"	"11"	"21"	"1:33.367"	"British Grand Prix"	"2008"
"Timo"	"Glock"	"12"	"19"	"1:34.610"	"British Grand Prix"	"2008"
"Felipe"	"Massa"	"13"	"19"	"1:33.257"	"British Grand Prix"	"2008"

"Robert"	"Kubica"	"\N"	"22"	"1:33.539"	"British Grand Prix"	"2008"
"Jenson"	"Button"	"\N"	"21"	"1:33.376"	"British Grand Prix"	"2008"
"Nelson"	"Piquet Jr."	"\N"	"21"	"1:33.203"	"British Grand Prix"	"2008"
"Giancarlo"	"Fisichella"	"\N"	"21"	"1:34.930"	"British Grand Prix"	"2008"
"Adrian"	"Sutil"	"\N"	"5"	"1:38.160"	"British Grand Prix"	"2008"
"Sebastian"	"Vettel"	"\N"	"\N"	"\N"	"British Grand Prix"	"2008"
"David"	"Coulthard"	"\N"	"\N"	"\N"	"British Grand Prix"	"2008"
"Lewis"	"Hamilton"	"1"	"17"	"1:16.039"	"German Grand Prix"	"2008"
"Nelson"	"Piquet Jr."	"2"	"66"	"1:16.910"	"German Grand Prix"	"2008"
"Felipe"	"Massa"	"3"	"19"	"1:16.502"	"German Grand Prix"	"2008"
"Nick"	"Heidfeld"	"4"	"52"	"1:15.987"	"German Grand Prix"	"2008"
"Heikki"	"Kovalainen"	"5"	"63"	"1:16.495"	"German Grand Prix"	"2008"
"Kimi"	"Räikkönen"	"6"	"66"	"1:16.342"	"German Grand Prix"	"2008"
"Robert"	"Kubica"	"7"	"17"	"1:16.610"	"German Grand Prix"	"2008"
"Sebastian"	"Vettel"	"8"	"20"	"1:16.772"	"German Grand Prix"	"2008"
"Jarno"	"Trulli"	"9"	"17"	"1:17.023"	"German Grand Prix"	"2008"
"Nico"	"Rosberg"	"10"	"34"	"1:17.380"	"German Grand Prix"	"2008"
"Fernando"	"Alonso"	"11"	"17"	"1:17.115"	"German Grand Prix"	"2008"
"Sébastien"	"Bourdais"	"12"	"63"	"1:16.969"	"German Grand Prix"	"2008"
"David"	"Coulthard"	"13"	"18"	"1:16.994"	"German Grand Prix"	"2008"
"Kazuki"	"Nakajima"	"14"	"23"	"1:17.691"	"German Grand Prix"	"2008"
"Adrian"	"Sutil"	"15"	"66"	"1:17.889"	"German Grand Prix"	"2008"
"Giancarlo"	"Fisichella"	"16"	"67"	"1:18.208"	"German Grand Prix"	"2008"

"Jenson"	"Button"	"17"	"56"	"1:17.636"	"German Grand Prix"	"2008"
"Rubens"	"Barrichello"	"\N"	"30"	"1:17.986"	"German Grand Prix"	"2008"
"Mark"	"Webber"	"\N"	"20"	"1:17.206"	"German Grand Prix"	"2008"
"Timo"	"Glock"	"\N"	"24"	"1:16.712"	"German Grand Prix"	"2008"
"Heikki"	"Kovalainen"	"1"	"19"	"1:21.753"	"Hungarian Grand Prix"	"2008"
"Timo"	"Glock"	"2"	"42"	"1:21.671"	"Hungarian Grand Prix"	"2008"
"Kimi"	"Räikkönen"	"3"	"61"	"1:21.195"	"Hungarian Grand Prix"	"2008"
"Fernando"	"Alonso"	"4"	"47"	"1:21.793"	"Hungarian Grand Prix"	"2008"
"Lewis"	"Hamilton"	"5"	"15"	"1:21.493"	"Hungarian Grand Prix"	"2008"
"Nelson"	"Piquet Jr."	"6"	"64"	"1:21.537"	"Hungarian Grand Prix"	"2008"
"Jarno"	"Trulli"	"7"	"65"	"1:21.638"	"Hungarian Grand Prix"	"2008"
"Robert"	"Kubica"	"8"	"70"	"1:21.941"	"Hungarian Grand Prix"	"2008"
"Mark"	"Webber"	"9"	"67"	"1:22.125"	"Hungarian Grand Prix"	"2008"
"Nick"	"Heidfeld"	"10"	"67"	"1:22.183"	"Hungarian Grand Prix"	"2008"
"David"	"Coulthard"	"11"	"67"	"1:22.732"	"Hungarian Grand Prix"	"2008"
"Jenson"	"Button"	"12"	"63"	"1:22.397"	"Hungarian Grand Prix"	"2008"
"Kazuki"	"Nakajima"	"13"	"67"	"1:23.307"	"Hungarian Grand Prix"	"2008"
"Nico"	"Rosberg"	"14"	"57"	"1:22.397"	"Hungarian Grand Prix"	"2008"
"Giancarlo"	"Fisichella"	"15"	"68"	"1:22.641"	"Hungarian Grand Prix"	"2008"
"Rubens"	"Barrichello"	"16"	"65"	"1:22.436"	"Hungarian Grand Prix"	"2008"
"Felipe"	"Massa"	"17"	"16"	"1:21.355"	"Hungarian Grand Prix"	"2008"
"Sébastien"	"Bourdais"	"18"	"29"	"1:23.220"	"Hungarian Grand Prix"	"2008"
"Adrian"	"Sutil"	"\N"	"29"	"1:23.650"	"Hungarian Grand Prix"	"2008"

"Sebastian"	"Vettel"	"\N"	"17"	"1:24.222"	"Hungarian Grand Prix"	"2008"
"Felipe"	"Massa"	"1"	"36"	"1:38.708"	"European Grand Prix"	"2008"
"Lewis"	"Hamilton"	"2"	"16"	"1:38.884"	"European Grand Prix"	"2008"
"Robert"	"Kubica"	"3"	"38"	"1:39.330"	"European Grand Prix"	"2008"
"Heikki"	"Kovalainen"	"4"	"19"	"1:39.112"	"European Grand Prix"	"2008"
"Jarno"	"Trulli"	"5"	"37"	"1:39.657"	"European Grand Prix"	"2008"
"Sebastian"	"Vettel"	"6"	"36"	"1:39.485"	"European Grand Prix"	"2008"
"Timo"	"Glock"	"7"	"54"	"1:39.535"	"European Grand Prix"	"2008"
"Nico"	"Rosberg"	"8"	"35"	"1:39.577"	"European Grand Prix"	"2008"
"Nick"	"Heidfeld"	"9"	"57"	"1:39.526"	"European Grand Prix"	"2008"
"Sébastien"	"Bourdais"	"10"	"35"	"1:39.639"	"European Grand Prix"	"2008"
"Nelson"	"Piquet Jr."	"11"	"29"	"1:39.544"	"European Grand Prix"	"2008"
"Mark"	"Webber"	"12"	"27"	"1:40.264"	"European Grand Prix"	"2008"
"Jenson"	"Button"	"13"	"21"	"1:40.763"	"European Grand Prix"	"2008"
"Giancarlo"	"Fisichella"	"14"	"54"	"1:40.353"	"European Grand Prix"	"2008"
"Kazuki"	"Nakajima"	"15"	"56"	"1:39.803"	"European Grand Prix"	"2008"
"Rubens"	"Barrichello"	"16"	"56"	"1:40.593"	"European Grand Prix"	"2008"
"David"	"Coulthard"	"17"	"13"	"1:40.978"	"European Grand Prix"	"2008"
"Kimi"	"Räikkönen"	"\N"	"34"	"1:39.424"	"European Grand Prix"	"2008"
"Adrian"	"Sutil"	"\N"	"28"	"1:40.661"	"European Grand Prix"	"2008"
"Fernando"	"Alonso"	"\N"	"\N"	"\N"	"European Grand Prix"	"2008"
"Felipe"	"Massa"	"1"	"26"	"1:48.222"	"Belgian Grand Prix"	"2008"
"Nick"	"Heidfeld"	"2"	"36"	"1:49.067"	"Belgian Grand Prix"	"2008"

"Lewis"	"Hamilton"	"3"	"20"	"1:48.135"	"Belgian Grand Prix"	"2008"
"Fernando"	"Alonso"	"4"	"25"	"1:49.238"	"Belgian Grand Prix"	"2008"
"Sebastian"	"Vettel"	"5"	"16"	"1:49.086"	"Belgian Grand Prix"	"2008"
"Robert"	"Kubica"	"6"	"36"	"1:48.965"	"Belgian Grand Prix"	"2008"
"Sébastien"	"Bourdais"	"7"	"31"	"1:49.002"	"Belgian Grand Prix"	"2008"
"Mark"	"Webber"	"8"	"34"	"1:49.515"	"Belgian Grand Prix"	"2008"
"Timo"	"Glock"	"9"	"37"	"1:50.255"	"Belgian Grand Prix"	"2008"
"Heikki"	"Kovalainen"	"10"	"35"	"1:48.223"	"Belgian Grand Prix"	"2008"
"David"	"Coulthard"	"11"	"19"	"1:50.177"	"Belgian Grand Prix"	"2008"
"Nico"	"Rosberg"	"12"	"32"	"1:50.656"	"Belgian Grand Prix"	"2008"
"Adrian"	"Sutil"	"13"	"27"	"1:50.487"	"Belgian Grand Prix"	"2008"
"Kazuki"	"Nakajima"	"14"	"35"	"1:50.970"	"Belgian Grand Prix"	"2008"
"Jenson"	"Button"	"15"	"19"	"1:50.671"	"Belgian Grand Prix"	"2008"
"Jarno"	"Trulli"	"16"	"32"	"1:50.543"	"Belgian Grand Prix"	"2008"
"Giancarlo"	"Fisichella"	"17"	"17"	"1:51.701"	"Belgian Grand Prix"	"2008"
"Kimi"	"Räikkönen"	"18"	"24"	"1:47.930"	"Belgian Grand Prix"	"2008"
"Rubens"	"Barrichello"	"\N"	"10"	"1:52.072"	"Belgian Grand Prix"	"2008"
"Nelson"	"Piquet Jr."	"\N"	"12"	"1:51.118"	"Belgian Grand Prix"	"2008"
"Sebastian"	"Vettel"	"1"	"53"	"1:30.510"	"Italian Grand Prix"	"2008"
"Heikki"	"Kovalainen"	"2"	"53"	"1:30.300"	"Italian Grand Prix"	"2008"
"Robert"	"Kubica"	"3"	"52"	"1:30.298"	"Italian Grand Prix"	"2008"
"Fernando"	"Alonso"	"4"	"51"	"1:29.961"	"Italian Grand Prix"	"2008"
"Nick"	"Heidfeld"	"5"	"53"	"1:29.807"	"Italian Grand Prix"	"2008"

"Felipe"	"Massa"	"6"	"52"	"1:29.696"	"Italian Grand Prix"	"2008"
"Lewis"	"Hamilton"	"7"	"52"	"1:29.721"	"Italian Grand Prix"	"2008"
"Mark"	"Webber"	"8"	"52"	"1:29.681"	"Italian Grand Prix"	"2008"
"Kimi"	"Räikkönen"	"9"	"53"	"1:28.047"	"Italian Grand Prix"	"2008"
"Nelson"	"Piquet Jr."	"10"	"53"	"1:30.918"	"Italian Grand Prix"	"2008"
"Timo"	"Glock"	"11"	"53"	"1:29.948"	"Italian Grand Prix"	"2008"
"Kazuki"	"Nakajima"	"12"	"53"	"1:30.215"	"Italian Grand Prix"	"2008"
"Jarno"	"Trulli"	"13"	"52"	"1:30.853"	"Italian Grand Prix"	"2008"
"Nico"	"Rosberg"	"14"	"53"	"1:30.019"	"Italian Grand Prix"	"2008"
"Jenson"	"Button"	"15"	"52"	"1:29.827"	"Italian Grand Prix"	"2008"
"David"	"Coulthard"	"16"	"49"	"1:32.459"	"Italian Grand Prix"	"2008"
"Rubens"	"Barrichello"	"17"	"42"	"1:33.918"	"Italian Grand Prix"	"2008"
"Sébastien"	"Bourdais"	"18"	"52"	"1:29.258"	"Italian Grand Prix"	"2008"
"Adrian"	"Sutil"	"19"	"51"	"1:33.458"	"Italian Grand Prix"	"2008"
"Giancarlo"	"Fisichella"	"\N"	"6"	"1:37.304"	"Italian Grand Prix"	"2008"
"Fernando"	"Alonso"	"1"	"55"	"1:45.768"	"Singapore Grand Prix"	"2008"
"Nico"	"Rosberg"	"2"	"39"	"1:46.454"	"Singapore Grand Prix"	"2008"
"Lewis"	"Hamilton"	"3"	"14"	"1:46.072"	"Singapore Grand Prix"	"2008"
"Timo"	"Glock"	"4"	"50"	"1:47.044"	"Singapore Grand Prix"	"2008"
"Sebastian"	"Vettel"	"5"	"13"	"1:47.271"	"Singapore Grand Prix"	"2008"
"Nick"	"Heidfeld"	"6"	"14"	"1:47.306"	"Singapore Grand Prix"	"2008"
"David"	"Coulthard"	"7"	"41"	"1:47.562"	"Singapore Grand Prix"	"2008"
"Kazuki"	"Nakajima"	"8"	"13"	"1:47.287"	"Singapore Grand Prix"	"2008"

"Jenson"	"Button"	"9"	"61"	"1:48.128"	"Singapore Grand Prix"	"2008"
"Heikki"	"Kovalainen"	"10"	"14"	"1:47.337"	"Singapore Grand Prix"	"2008"
"Robert"	"Kubica"	"11"	"14"	"1:46.899"	"Singapore Grand Prix"	"2008"
"Sébastien"	"Bourdais"	"12"	"29"	"1:47.820"	"Singapore Grand Prix"	"2008"
"Felipe"	"Massa"	"13"	"13"	"1:45.757"	"Singapore Grand Prix"	"2008"
"Giancarlo"	"Fisichella"	"14"	"28"	"1:49.101"	"Singapore Grand Prix"	"2008"
"Kimi"	"Räikkönen"	"15"	"14"	"1:45.599"	"Singapore Grand Prix"	"2008"
"Jarno"	"Trulli"	"\N"	"32"	"1:46.972"	"Singapore Grand Prix"	"2008"
"Adrian"	"Sutil"	"\N"	"38"	"1:49.270"	"Singapore Grand Prix"	"2008"
"Mark"	"Webber"	"\N"	"13"	"1:49.183"	"Singapore Grand Prix"	"2008"
"Rubens"	"Barrichello"	"\N"	"13"	"1:50.320"	"Singapore Grand Prix"	"2008"
"Nelson"	"Piquet Jr."	"\N"	"13"	"1:50.449"	"Singapore Grand Prix"	"2008"
"Fernando"	"Alonso"	"1"	"41"	"1:19.101"	"Japanese Grand Prix"	"2008"
"Robert"	"Kubica"	"2"	"16"	"1:19.292"	"Japanese Grand Prix"	"2008"
"Kimi"	"Räikkönen"	"3"	"66"	"1:18.995"	"Japanese Grand Prix"	"2008"
"Nelson"	"Piquet Jr."	"4"	"54"	"1:19.199"	"Japanese Grand Prix"	"2008"
"Jarno"	"Trulli"	"5"	"18"	"1:19.524"	"Japanese Grand Prix"	"2008"
"Sebastian"	"Vettel"	"6"	"21"	"1:19.617"	"Japanese Grand Prix"	"2008"
"Felipe"	"Massa"	"7"	"55"	"1:18.426"	"Japanese Grand Prix"	"2008"
"Mark"	"Webber"	"8"	"30"	"1:19.820"	"Japanese Grand Prix"	"2008"
"Nick"	"Heidfeld"	"9"	"67"	"1:19.461"	"Japanese Grand Prix"	"2008"
"Sébastien"	"Bourdais"	"10"	"49"	"1:19.262"	"Japanese Grand Prix"	"2008"
"Nico"	"Rosberg"	"11"	"66"	"1:19.531"	"Japanese Grand Prix"	"2008"

"Lewis"	"Hamilton"	"12"	"65"	"1:19.560"	"Japanese Grand Prix"	"2008"
"Rubens"	"Barrichello"	"13"	"66"	"1:20.575"	"Japanese Grand Prix"	"2008"
"Jenson"	"Button"	"14"	"66"	"1:20.849"	"Japanese Grand Prix"	"2008"
"Kazuki"	"Nakajima"	"15"	"32"	"1:20.364"	"Japanese Grand Prix"	"2008"
"Giancarlo"	"Fisichella"	"\N"	"20"	"1:21.577"	"Japanese Grand Prix"	"2008"
"Heikki"	"Kovalainen"	"\N"	"12"	"1:19.258"	"Japanese Grand Prix"	"2008"
"Adrian"	"Sutil"	"\N"	"6"	"1:21.189"	"Japanese Grand Prix"	"2008"
"Timo"	"Glock"	"\N"	"4"	"1:20.254"	"Japanese Grand Prix"	"2008"
"David"	"Coulthard"	"\N"	"\N"	"\N"	"Japanese Grand Prix"	"2008"
"Lewis"	"Hamilton"	"1"	"13"	"1:36.325"	"Chinese Grand Prix"	"2008"
"Felipe"	"Massa"	"2"	"36"	"1:36.591"	"Chinese Grand Prix"	"2008"
"Kimi"	"Räikkönen"	"3"	"14"	"1:36.483"	"Chinese Grand Prix"	"2008"
"Fernando"	"Alonso"	"4"	"56"	"1:36.659"	"Chinese Grand Prix"	"2008"
"Nick"	"Heidfeld"	"5"	"56"	"1:36.498"	"Chinese Grand Prix"	"2008"
"Robert"	"Kubica"	"6"	"56"	"1:36.854"	"Chinese Grand Prix"	"2008"
"Timo"	"Glock"	"7"	"56"	"1:36.727"	"Chinese Grand Prix"	"2008"
"Nelson"	"Piquet Jr."	"8"	"56"	"1:36.996"	"Chinese Grand Prix"	"2008"
"Sebastian"	"Vettel"	"9"	"55"	"1:37.212"	"Chinese Grand Prix"	"2008"
"David"	"Coulthard"	"10"	"27"	"1:37.753"	"Chinese Grand Prix"	"2008"
"Rubens"	"Barrichello"	"11"	"56"	"1:37.845"	"Chinese Grand Prix"	"2008"
"Kazuki"	"Nakajima"	"12"	"54"	"1:38.019"	"Chinese Grand Prix"	"2008"
"Sébastien"	"Bourdais"	"13"	"36"	"1:37.452"	"Chinese Grand Prix"	"2008"
"Mark"	"Webber"	"14"	"10"	"1:37.680"	"Chinese Grand Prix"	"2008"

"Nico"	"Rosberg"	"15"	"38"	"1:37.246"	"Chinese Grand Prix"	"2008"
"Jenson"	"Button"	"16"	"54"	"1:37.773"	"Chinese Grand Prix"	"2008"
"Giancarlo"	"Fisichella"	"17"	"54"	"1:38.372"	"Chinese Grand Prix"	"2008"
"Heikki"	"Kovalainen"	"\N"	"32"	"1:37.302"	"Chinese Grand Prix"	"2008"
"Adrian"	"Sutil"	"\N"	"13"	"1:39.683"	"Chinese Grand Prix"	"2008"
"Jarno"	"Trulli"	"\N"	"2"	"2:39.612"	"Chinese Grand Prix"	"2008"
"Felipe"	"Massa"	"1"	"36"	"1:13.736"	"Brazilian Grand Prix"	"2008"
"Fernando"	"Alonso"	"2"	"39"	"1:14.229"	"Brazilian Grand Prix"	"2008"
"Kimi"	"Räikkönen"	"3"	"59"	"1:14.117"	"Brazilian Grand Prix"	"2008"
"Sebastian"	"Vettel"	"4"	"25"	"1:14.214"	"Brazilian Grand Prix"	"2008"
"Lewis"	"Hamilton"	"5"	"31"	"1:14.159"	"Brazilian Grand Prix"	"2008"
"Timo"	"Glock"	"6"	"32"	"1:14.057"	"Brazilian Grand Prix"	"2008"
"Heikki"	"Kovalainen"	"7"	"36"	"1:14.207"	"Brazilian Grand Prix"	"2008"
"Jarno"	"Trulli"	"8"	"40"	"1:14.167"	"Brazilian Grand Prix"	"2008"
"Mark"	"Webber"	"9"	"40"	"1:15.033"	"Brazilian Grand Prix"	"2008"
"Nick"	"Heidfeld"	"10"	"41"	"1:14.652"	"Brazilian Grand Prix"	"2008"
"Robert"	"Kubica"	"11"	"61"	"1:14.375"	"Brazilian Grand Prix"	"2008"
"Nico"	"Rosberg"	"12"	"35"	"1:14.934"	"Brazilian Grand Prix"	"2008"
"Jenson"	"Button"	"13"	"59"	"1:14.759"	"Brazilian Grand Prix"	"2008"
"Sébastien"	"Bourdais"	"14"	"57"	"1:14.951"	"Brazilian Grand Prix"	"2008"
"Rubens"	"Barrichello"	"15"	"35"	"1:15.414"	"Brazilian Grand Prix"	"2008"
"Adrian"	"Sutil"	"16"	"60"	"1:15.773"	"Brazilian Grand Prix"	"2008"
"Kazuki"	"Nakajima"	"17"	"50"	"1:15.865"	"Brazilian Grand Prix"	"2008"

"Giancarlo"	"Fisichella"	"18"	"36"	"1:15.212"	"Brazilian Grand Prix"	"2008"
"Nelson"	"Piquet Jr."	"\N"	"\N"	"\N"	"Brazilian Grand Prix"	"2008"
"David"	"Coulthard"	"\N"	"\N"	"\N"	"Brazilian Grand Prix"	"2008"
"Kimi"	"Räikkönen"	"1"	"41"	"1:25.235"	"Australian Grand Prix"	"2007"
"Fernando"	"Alonso"	"2"	"20"	"1:26.314"	"Australian Grand Prix"	"2007"
"Lewis"	"Hamilton"	"3"	"20"	"1:26.351"	"Australian Grand Prix"	"2007"
"Nick"	"Heidfeld"	"4"	"37"	"1:26.722"	"Australian Grand Prix"	"2007"
"Giancarlo"	"Fisichella"	"5"	"18"	"1:26.892"	"Australian Grand Prix"	"2007"
"Felipe"	"Massa"	"6"	"28"	"1:27.044"	"Australian Grand Prix"	"2007"
"Nico"	"Rosberg"	"7"	"40"	"1:26.721"	"Australian Grand Prix"	"2007"
"Ralf"	"Schumacher"	"8"	"42"	"1:27.796"	"Australian Grand Prix"	"2007"
"Jarno"	"Trulli"	"9"	"45"	"1:28.034"	"Australian Grand Prix"	"2007"
"Heikki"	"Kovalainen"	"10"	"44"	"1:27.592"	"Australian Grand Prix"	"2007"
"Rubens"	"Barrichello"	"11"	"17"	"1:28.098"	"Australian Grand Prix"	"2007"
"Takuma"	"Sato"	"12"	"20"	"1:28.487"	"Australian Grand Prix"	"2007"
"Mark"	"Webber"	"13"	"21"	"1:27.501"	"Australian Grand Prix"	"2007"
"Vitantonio"	"Liuzzi"	"14"	"44"	"1:28.282"	"Australian Grand Prix"	"2007"
"Jenson"	"Button"	"15"	"42"	"1:28.387"	"Australian Grand Prix"	"2007"
"Anthony"	"Davidson"	"16"	"41"	"1:28.489"	"Australian Grand Prix"	"2007"
"Adrian"	"Sutil"	"17"	"40"	"1:28.687"	"Australian Grand Prix"	"2007"
"Alexander"	"Wurz"	"\N"	"24"	"1:28.303"	"Australian Grand Prix"	"2007"
"David"	"Coulthard"	"\N"	"44"	"1:27.706"	"Australian Grand Prix"	"2007"
"Robert"	"Kubica"	"\N"	"19"	"1:26.642"	"Australian Grand Prix"	"2007"

"Scott"	"Speed"	"\N"	"22"	"1:28.953"	"Australian Grand Prix"	"2007"
"Christijan"	"Albers"	"\N"	"9"	"1:30.899"	"Australian Grand Prix"	"2007"
"Fernando"	"Alonso"	"1"	"42"	"1:36.861"	"Malaysian Grand Prix"	"2007"
"Lewis"	"Hamilton"	"2"	"22"	"1:36.701"	"Malaysian Grand Prix"	"2007"
"Kimi"	"Räikkönen"	"3"	"40"	"1:37.228"	"Malaysian Grand Prix"	"2007"
"Nick"	"Heidfeld"	"4"	"55"	"1:37.417"	"Malaysian Grand Prix"	"2007"
"Felipe"	"Massa"	"5"	"42"	"1:37.199"	"Malaysian Grand Prix"	"2007"
"Giancarlo"	"Fisichella"	"6"	"44"	"1:37.879"	"Malaysian Grand Prix"	"2007"
"Jarno"	"Trulli"	"7"	"40"	"1:38.016"	"Malaysian Grand Prix"	"2007"
"Heikki"	"Kovalainen"	"8"	"41"	"1:37.810"	"Malaysian Grand Prix"	"2007"
"Alexander"	"Wurz"	"9"	"19"	"1:37.864"	"Malaysian Grand Prix"	"2007"
"Mark"	"Webber"	"10"	"55"	"1:38.540"	"Malaysian Grand Prix"	"2007"
"Rubens"	"Barrichello"	"11"	"32"	"1:38.566"	"Malaysian Grand Prix"	"2007"
"Jenson"	"Button"	"12"	"54"	"1:38.658"	"Malaysian Grand Prix"	"2007"
"Takuma"	"Sato"	"13"	"37"	"1:38.496"	"Malaysian Grand Prix"	"2007"
"Scott"	"Speed"	"14"	"54"	"1:39.098"	"Malaysian Grand Prix"	"2007"
"Ralf"	"Schumacher"	"15"	"52"	"1:39.243"	"Malaysian Grand Prix"	"2007"
"Anthony"	"Davidson"	"16"	"32"	"1:39.566"	"Malaysian Grand Prix"	"2007"
"Vitantonio"	"Liuzzi"	"17"	"21"	"1:38.447"	"Malaysian Grand Prix"	"2007"
"Robert"	"Kubica"	"18"	"53"	"1:38.874"	"Malaysian Grand Prix"	"2007"
"Nico"	"Rosberg"	"\N"	"18"	"1:37.704"	"Malaysian Grand Prix"	"2007"
"David"	"Coulthard"	"\N"	"28"	"1:38.098"	"Malaysian Grand Prix"	"2007"
"Christijan"	"Albers"	"\N"	"5"	"1:41.495"	"Malaysian Grand Prix"	"2007"

"Adrian"	"Sutil"	"\N"	"\N"	"\N"	"Malaysian Grand Prix"	"2007"
"Felipe"	"Massa"	"1"	"42"	"1:34.067"	"Bahrain Grand Prix"	"2007"
"Lewis"	"Hamilton"	"2"	"18"	"1:34.270"	"Bahrain Grand Prix"	"2007"
"Kimi"	"Räikkönen"	"3"	"39"	"1:34.357"	"Bahrain Grand Prix"	"2007"
"Nick"	"Heidfeld"	"4"	"38"	"1:34.470"	"Bahrain Grand Prix"	"2007"
"Fernando"	"Alonso"	"5"	"46"	"1:34.420"	"Bahrain Grand Prix"	"2007"
"Robert"	"Kubica"	"6"	"46"	"1:34.819"	"Bahrain Grand Prix"	"2007"
"Jarno"	"Trulli"	"7"	"40"	"1:35.153"	"Bahrain Grand Prix"	"2007"
"Giancarlo"	"Fisichella"	"8"	"51"	"1:35.200"	"Bahrain Grand Prix"	"2007"
"Heikki"	"Kovalainen"	"9"	"27"	"1:35.475"	"Bahrain Grand Prix"	"2007"
"Nico"	"Rosberg"	"10"	"51"	"1:35.556"	"Bahrain Grand Prix"	"2007"
"Alexander"	"Wurz"	"11"	"27"	"1:35.992"	"Bahrain Grand Prix"	"2007"
"Ralf"	"Schumacher"	"12"	"37"	"1:35.845"	"Bahrain Grand Prix"	"2007"
"Rubens"	"Barrichello"	"13"	"50"	"1:35.842"	"Bahrain Grand Prix"	"2007"
"Christijan"	"Albers"	"14"	"40"	"1:37.184"	"Bahrain Grand Prix"	"2007"
"Adrian"	"Sutil"	"15"	"45"	"1:36.772"	"Bahrain Grand Prix"	"2007"
"Anthony"	"Davidson"	"16"	"27"	"1:36.111"	"Bahrain Grand Prix"	"2007"
"Mark"	"Webber"	"\N"	"37"	"1:35.705"	"Bahrain Grand Prix"	"2007"
"David"	"Coulthard"	"\N"	"34"	"1:35.384"	"Bahrain Grand Prix"	"2007"
"Takuma"	"Sato"	"\N"	"23"	"1:36.359"	"Bahrain Grand Prix"	"2007"
"Vitantonio"	"Liuzzi"	"\N"	"23"	"1:35.723"	"Bahrain Grand Prix"	"2007"
"Jenson"	"Button"	"\N"	"\N"	"\N"	"Bahrain Grand Prix"	"2007"
"Scott"	"Speed"	"\N"	"\N"	"\N"	"Bahrain Grand Prix"	"2007"

"Felipe"	"Massa"	"1"	"14"	"1:22.680"	"Spanish Grand Prix"	"2007"
"Lewis"	"Hamilton"	"2"	"20"	"1:22.876"	"Spanish Grand Prix"	"2007"
"Fernando"	"Alonso"	"3"	"17"	"1:22.966"	"Spanish Grand Prix"	"2007"
"Robert"	"Kubica"	"4"	"20"	"1:23.129"	"Spanish Grand Prix"	"2007"
"David"	"Coulthard"	"5"	"18"	"1:23.524"	"Spanish Grand Prix"	"2007"
"Nico"	"Rosberg"	"6"	"60"	"1:23.693"	"Spanish Grand Prix"	"2007"
"Heikki"	"Kovalainen"	"7"	"32"	"1:22.980"	"Spanish Grand Prix"	"2007"
"Takuma"	"Sato"	"8"	"23"	"1:24.110"	"Spanish Grand Prix"	"2007"
"Giancarlo"	"Fisichella"	"9"	"57"	"1:23.560"	"Spanish Grand Prix"	"2007"
"Rubens"	"Barrichello"	"10"	"16"	"1:24.287"	"Spanish Grand Prix"	"2007"
"Anthony"	"Davidson"	"11"	"59"	"1:24.291"	"Spanish Grand Prix"	"2007"
"Jenson"	"Button"	"12"	"64"	"1:24.186"	"Spanish Grand Prix"	"2007"
"Adrian"	"Sutil"	"13"	"57"	"1:25.191"	"Spanish Grand Prix"	"2007"
"Christijan"	"Albers"	"14"	"61"	"1:25.260"	"Spanish Grand Prix"	"2007"
"Nick"	"Heidfeld"	"\N"	"22"	"1:23.483"	"Spanish Grand Prix"	"2007"
"Ralf"	"Schumacher"	"\N"	"37"	"1:24.003"	"Spanish Grand Prix"	"2007"
"Vitantonio"	"Liuzzi"	"\N"	"18"	"1:25.207"	"Spanish Grand Prix"	"2007"
"Scott"	"Speed"	"\N"	"6"	"1:26.238"	"Spanish Grand Prix"	"2007"
"Kimi"	"Räikkönen"	"\N"	"7"	"1:23.475"	"Spanish Grand Prix"	"2007"
"Jarno"	"Trulli"	"\N"	"6"	"1:26.094"	"Spanish Grand Prix"	"2007"
"Mark"	"Webber"	"\N"	"4"	"1:26.323"	"Spanish Grand Prix"	"2007"
"Alexander"	"Wurz"	"\N"	"\N"	"\N"	"Spanish Grand Prix"	"2007"
"Fernando"	"Alonso"	"1"	"44"	"1:15.284"	"Monaco Grand Prix"	"2007"

"Lewis"	"Hamilton"	"2"	"28"	"1:15.372"	"Monaco Grand Prix"	"2007"
"Felipe"	"Massa"	"3"	"47"	"1:16.183"	"Monaco Grand Prix"	"2007"
"Giancarlo"	"Fisichella"	"4"	"54"	"1:16.254"	"Monaco Grand Prix"	"2007"
"Robert"	"Kubica"	"5"	"39"	"1:16.006"	"Monaco Grand Prix"	"2007"
"Nick"	"Heidfeld"	"6"	"30"	"1:17.041"	"Monaco Grand Prix"	"2007"
"Alexander"	"Wurz"	"7"	"40"	"1:16.658"	"Monaco Grand Prix"	"2007"
"Kimi"	"Räikkönen"	"8"	"62"	"1:16.592"	"Monaco Grand Prix"	"2007"
"Scott"	"Speed"	"9"	"73"	"1:16.867"	"Monaco Grand Prix"	"2007"
"Rubens"	"Barrichello"	"10"	"69"	"1:17.080"	"Monaco Grand Prix"	"2007"
"Jenson"	"Button"	"11"	"40"	"1:16.802"	"Monaco Grand Prix"	"2007"
"Nico"	"Rosberg"	"12"	"70"	"1:16.991"	"Monaco Grand Prix"	"2007"
"Heikki"	"Kovalainen"	"13"	"72"	"1:17.100"	"Monaco Grand Prix"	"2007"
"David"	"Coulthard"	"14"	"75"	"1:16.786"	"Monaco Grand Prix"	"2007"
"Jarno"	"Trulli"	"15"	"53"	"1:17.495"	"Monaco Grand Prix"	"2007"
"Ralf"	"Schumacher"	"16"	"47"	"1:17.231"	"Monaco Grand Prix"	"2007"
"Takuma"	"Sato"	"17"	"74"	"1:17.183"	"Monaco Grand Prix"	"2007"
"Anthony"	"Davidson"	"18"	"63"	"1:17.223"	"Monaco Grand Prix"	"2007"
"Christijan"	"Albers"	"19"	"70"	"1:17.689"	"Monaco Grand Prix"	"2007"
"Adrian"	"Sutil"	"\N"	"34"	"1:17.678"	"Monaco Grand Prix"	"2007"
"Mark"	"Webber"	"\N"	"17"	"1:18.998"	"Monaco Grand Prix"	"2007"
"Vitantonio"	"Liuzzi"	"\N"	"\N"	"\N"	"Monaco Grand Prix"	"2007"
"Lewis"	"Hamilton"	"1"	"37"	"1:16.494"	"Canadian Grand Prix"	"2007"
"Nick"	"Heidfeld"	"2"	"19"	"1:16.696"	"Canadian Grand Prix"	"2007"

"Alexander"	"Wurz"	"3"	"67"	"1:17.947"	"Canadian Grand Prix"	"2007"
"Heikki"	"Kovalainen"	"4"	"67"	"1:18.368"	"Canadian Grand Prix"	"2007"
"Kimi"	"Räikkönen"	"5"	"21"	"1:16.861"	"Canadian Grand Prix"	"2007"
"Takuma"	"Sato"	"6"	"47"	"1:18.035"	"Canadian Grand Prix"	"2007"
"Fernando"	"Alonso"	"7"	"46"	"1:16.367"	"Canadian Grand Prix"	"2007"
"Ralf"	"Schumacher"	"8"	"38"	"1:17.910"	"Canadian Grand Prix"	"2007"
"Mark"	"Webber"	"9"	"47"	"1:17.618"	"Canadian Grand Prix"	"2007"
"Nico"	"Rosberg"	"10"	"42"	"1:17.156"	"Canadian Grand Prix"	"2007"
"Anthony"	"Davidson"	"11"	"36"	"1:18.780"	"Canadian Grand Prix"	"2007"
"Rubens"	"Barrichello"	"12"	"62"	"1:18.543"	"Canadian Grand Prix"	"2007"
"Jarno"	"Trulli"	"\N"	"12"	"1:19.092"	"Canadian Grand Prix"	"2007"
"Vitantonio"	"Liuzzi"	"\N"	"9"	"1:19.375"	"Canadian Grand Prix"	"2007"
"Christijan"	"Albers"	"\N"	"21"	"1:19.254"	"Canadian Grand Prix"	"2007"
"David"	"Coulthard"	"\N"	"19"	"1:18.981"	"Canadian Grand Prix"	"2007"
"Robert"	"Kubica"	"\N"	"19"	"1:17.529"	"Canadian Grand Prix"	"2007"
"Adrian"	"Sutil"	"\N"	"20"	"1:19.452"	"Canadian Grand Prix"	"2007"
"Scott"	"Speed"	"\N"	"6"	"1:20.092"	"Canadian Grand Prix"	"2007"
"Jenson"	"Button"	"\N"	"\N"	"\N"	"Canadian Grand Prix"	"2007"
"Felipe"	"Massa"	"\N"	"22"	"1:16.849"	"Canadian Grand Prix"	"2007"
"Giancarlo"	"Fisichella"	"\N"	"22"	"1:17.411"	"Canadian Grand Prix"	"2007"
"Lewis"	"Hamilton"	"1"	"20"	"1:13.222"	"United States Grand Prix"	"2007"
"Fernando"	"Alonso"	"2"	"21"	"1:13.257"	"United States Grand Prix"	"2007"
"Felipe"	"Massa"	"3"	"50"	"1:13.380"	"United States Grand Prix"	"2007"

"Kimi"	"Räikkönen"	"4"	"49"	"1:13.117"	"United States Grand Prix"	"2007"
"Heikki"	"Kovalainen"	"5"	"67"	"1:13.998"	"United States Grand Prix"	"2007"
"Jarno"	"Trulli"	"6"	"30"	"1:14.016"	"United States Grand Prix"	"2007"
"Mark"	"Webber"	"7"	"57"	"1:14.004"	"United States Grand Prix"	"2007"
"Sebastian"	"Vettel"	"8"	"53"	"1:13.862"	"United States Grand Prix"	"2007"
"Giancarlo"	"Fisichella"	"9"	"67"	"1:14.009"	"United States Grand Prix"	"2007"
"Alexander"	"Wurz"	"10"	"41"	"1:14.486"	"United States Grand Prix"	"2007"

Appendix I: Output of Query to Return Lap Number of Race Winner

```
-------------------------------------------------------------------------+
| forename   | surname     | pt.lap | r.name                | r.year |
+-------------------------------------------------------------------------+
| "Sebastian" | "Vettel"    | "14"   | "Australian Grand Prix"   | "2011" |
| "Sebastian" | "Vettel"    | "36"   | "Australian Grand Prix"   | "2011" |
| "Sebastian" | "Vettel"    | "13"   | "Malaysian Grand Prix"    | "2011" |
| "Sebastian" | "Vettel"    | "25"   | "Malaysian Grand Prix"    | "2011" |
| "Sebastian" | "Vettel"    | "41"   | "Malaysian Grand Prix"    | "2011" |
| "Lewis"    | "Hamilton"  | "15"   | "Chinese Grand Prix"      | "2011" |
| "Lewis"    | "Hamilton"  | "25"   | "Chinese Grand Prix"      | "2011" |
| "Lewis"    | "Hamilton"  | "38"   | "Chinese Grand Prix"      | "2011" |
| "Sebastian" | "Vettel"    | "11"   | "Turkish Grand Prix"      | "2011" |
| "Sebastian" | "Vettel"    | "25"   | "Turkish Grand Prix"      | "2011" |
| "Sebastian" | "Vettel"    | "40"   | "Turkish Grand Prix"      | "2011" |
| "Sebastian" | "Vettel"    | "47"   | "Turkish Grand Prix"      | "2011" |
| "Sebastian" | "Vettel"    | "9"    | "Spanish Grand Prix"      | "2011" |
| "Sebastian" | "Vettel"    | "18"   | "Spanish Grand Prix"      | "2011" |
| "Sebastian" | "Vettel"    | "34"   | "Spanish Grand Prix"      | "2011" |
| "Sebastian" | "Vettel"    | "48"   | "Spanish Grand Prix"      | "2011" |
| "Sebastian" | "Vettel"    | "16"   | "Monaco Grand Prix"       | "2011" |
| "Jenson"   | "Button"    | "8"    | "Canadian Grand Prix"     | "2011" |
| "Jenson"   | "Button"    | "13"   | "Canadian Grand Prix"     | "2011" |
| "Jenson"   | "Button"    | "19"   | "Canadian Grand Prix"     | "2011" |
| "Jenson"   | "Button"    | "35"   | "Canadian Grand Prix"     | "2011" |
| "Jenson"   | "Button"    | "37"   | "Canadian Grand Prix"     | "2011" |
| "Jenson"   | "Button"    | "51"   | "Canadian Grand Prix"     | "2011" |
| "Sebastian" | "Vettel"    | "14"   | "European Grand Prix"     | "2011" |
| "Sebastian" | "Vettel"    | "30"   | "European Grand Prix"     | "2011" |
| "Sebastian" | "Vettel"    | "47"   | "European Grand Prix"     | "2011" |
| "Fernando"  | "Alonso"    | "12"   | "British Grand Prix"      | "2011" |
| "Fernando"  | "Alonso"    | "27"   | "British Grand Prix"      | "2011" |
| "Fernando"  | "Alonso"    | "39"   | "British Grand Prix"      | "2011" |
| "Lewis"    | "Hamilton"  | "16"   | "German Grand Prix"       | "2011" |
| "Lewis"    | "Hamilton"  | "31"   | "German Grand Prix"       | "2011" |
| "Lewis"    | "Hamilton"  | "51"   | "German Grand Prix"       | "2011" |
| "Jenson"   | "Button"    | "11"   | "Hungarian Grand Prix"    | "2011" |
| "Jenson"   | "Button"    | "27"   | "Hungarian Grand Prix"    | "2011" |
| "Jenson"   | "Button"    | "42"   | "Hungarian Grand Prix"    | "2011" |
| "Sebastian" | "Vettel"    | "5"    | "Belgian Grand Prix"      | "2011" |
| "Sebastian" | "Vettel"    | "13"   | "Belgian Grand Prix"      | "2011" |
| "Sebastian" | "Vettel"    | "30"   | "Belgian Grand Prix"      | "2011" |
| "Sebastian" | "Vettel"    | "20"   | "Italian Grand Prix"      | "2011" |
| "Sebastian" | "Vettel"    | "35"   | "Italian Grand Prix"      | "2011" |
| "Sebastian" | "Vettel"    | "14"   | "Singapore Grand Prix"    | "2011" |
| "Sebastian" | "Vettel"    | "30"   | "Singapore Grand Prix"    | "2011" |
```

"Sebastian"	"Vettel"	"49"	"Singapore Grand Prix"	"2011"
"Jenson"	"Button"	"10"	"Japanese Grand Prix"	"2011"
"Jenson"	"Button"	"20"	"Japanese Grand Prix"	"2011"
"Jenson"	"Button"	"36"	"Japanese Grand Prix"	"2011"
"Sebastian"	"Vettel"	"16"	"Korean Grand Prix"	"2011"
"Sebastian"	"Vettel"	"34"	"Korean Grand Prix"	"2011"
"Sebastian"	"Vettel"	"19"	"Indian Grand Prix"	"2011"
"Sebastian"	"Vettel"	"47"	"Indian Grand Prix"	"2011"
"Lewis"	"Hamilton"	"16"	"Abu Dhabi Grand Prix"	"2011"
"Lewis"	"Hamilton"	"40"	"Abu Dhabi Grand Prix"	"2011"
"Mark"	"Webber"	"18"	"Brazilian Grand Prix"	"2011"
"Mark"	"Webber"	"38"	"Brazilian Grand Prix"	"2011"
"Mark"	"Webber"	"58"	"Brazilian Grand Prix"	"2011"
"Jenson"	"Button"	"16"	"Australian Grand Prix"	"2012"
"Jenson"	"Button"	"36"	"Australian Grand Prix"	"2012"
"Fernando"	"Alonso"	"4"	"Malaysian Grand Prix"	"2012"
"Fernando"	"Alonso"	"14"	"Malaysian Grand Prix"	"2012"
"Fernando"	"Alonso"	"40"	"Malaysian Grand Prix"	"2012"
"Nico"	"Rosberg"	"34"	"Chinese Grand Prix"	"2012"
"Nico"	"Rosberg"	"13"	"Chinese Grand Prix"	"2012"
"Sebastian"	"Vettel"	"11"	"Bahrain Grand Prix"	"2012"
"Sebastian"	"Vettel"	"25"	"Bahrain Grand Prix"	"2012"
"Sebastian"	"Vettel"	"39"	"Bahrain Grand Prix"	"2012"
"Pastor"	"Maldonado"	"11"	"Spanish Grand Prix"	"2012"
"Pastor"	"Maldonado"	"24"	"Spanish Grand Prix"	"2012"
"Pastor"	"Maldonado"	"41"	"Spanish Grand Prix"	"2012"
"Mark"	"Webber"	"29"	"Monaco Grand Prix"	"2012"
"Lewis"	"Hamilton"	"17"	"Canadian Grand Prix"	"2012"
"Lewis"	"Hamilton"	"50"	"Canadian Grand Prix"	"2012"
"Fernando"	"Alonso"	"15"	"European Grand Prix"	"2012"
"Fernando"	"Alonso"	"28"	"European Grand Prix"	"2012"
"Mark"	"Webber"	"33"	"British Grand Prix"	"2012"
"Mark"	"Webber"	"14"	"British Grand Prix"	"2012"
"Fernando"	"Alonso"	"18"	"German Grand Prix"	"2012"
"Fernando"	"Alonso"	"41"	"German Grand Prix"	"2012"
"Lewis"	"Hamilton"	"18"	"Hungarian Grand Prix"	"2012"
"Lewis"	"Hamilton"	"40"	"Hungarian Grand Prix"	"2012"
"Jenson"	"Button"	"20"	"Belgian Grand Prix"	"2012"
"Lewis"	"Hamilton"	"23"	"Italian Grand Prix"	"2012"
"Sebastian"	"Vettel"	"33"	"Singapore Grand Prix"	"2012"
"Sebastian"	"Vettel"	"10"	"Singapore Grand Prix"	"2012"
"Sebastian"	"Vettel"	"17"	"Japanese Grand Prix"	"2012"
"Sebastian"	"Vettel"	"37"	"Japanese Grand Prix"	"2012"
"Sebastian"	"Vettel"	"15"	"Korean Grand Prix"	"2012"
"Sebastian"	"Vettel"	"35"	"Korean Grand Prix"	"2012"
"Sebastian"	"Vettel"	"33"	"Indian Grand Prix"	"2012"
"Kimi"	"Räikkönen"	"31"	"Abu Dhabi Grand Prix"	"2012"

"Lewis"	"Hamilton"	"20"	"United States Grand Prix"	"2012"
"Jenson"	"Button"	"23"	"Brazilian Grand Prix"	"2012"
"Jenson"	"Button"	"57"	"Brazilian Grand Prix"	"2012"
"Kimi"	"Räikkönen"	"9"	"Australian Grand Prix"	"2013"
"Kimi"	"Räikkönen"	"34"	"Australian Grand Prix"	"2013"
"Sebastian"	"Vettel"	"5"	"Malaysian Grand Prix"	"2013"
"Sebastian"	"Vettel"	"22"	"Malaysian Grand Prix"	"2013"
"Sebastian"	"Vettel"	"32"	"Malaysian Grand Prix"	"2013"
"Sebastian"	"Vettel"	"42"	"Malaysian Grand Prix"	"2013"
"Fernando"	"Alonso"	"6"	"Chinese Grand Prix"	"2013"
"Fernando"	"Alonso"	"23"	"Chinese Grand Prix"	"2013"
"Fernando"	"Alonso"	"41"	"Chinese Grand Prix"	"2013"
"Sebastian"	"Vettel"	"10"	"Bahrain Grand Prix"	"2013"
"Sebastian"	"Vettel"	"25"	"Bahrain Grand Prix"	"2013"
"Sebastian"	"Vettel"	"42"	"Bahrain Grand Prix"	"2013"
"Fernando"	"Alonso"	"9"	"Spanish Grand Prix"	"2013"
"Fernando"	"Alonso"	"21"	"Spanish Grand Prix"	"2013"
"Fernando"	"Alonso"	"36"	"Spanish Grand Prix"	"2013"
"Fernando"	"Alonso"	"49"	"Spanish Grand Prix"	"2013"
"Nico"	"Rosberg"	"31"	"Monaco Grand Prix"	"2013"
"Sebastian"	"Vettel"	"16"	"Canadian Grand Prix"	"2013"
"Sebastian"	"Vettel"	"49"	"Canadian Grand Prix"	"2013"
"Nico"	"Rosberg"	"12"	"British Grand Prix"	"2013"
"Nico"	"Rosberg"	"34"	"British Grand Prix"	"2013"
"Nico"	"Rosberg"	"42"	"British Grand Prix"	"2013"
"Sebastian"	"Vettel"	"7"	"German Grand Prix"	"2013"
"Sebastian"	"Vettel"	"24"	"German Grand Prix"	"2013"
"Sebastian"	"Vettel"	"41"	"German Grand Prix"	"2013"
"Lewis"	"Hamilton"	"9"	"Hungarian Grand Prix"	"2013"
"Lewis"	"Hamilton"	"31"	"Hungarian Grand Prix"	"2013"
"Lewis"	"Hamilton"	"50"	"Hungarian Grand Prix"	"2013"
"Sebastian"	"Vettel"	"14"	"Belgian Grand Prix"	"2013"
"Sebastian"	"Vettel"	"30"	"Belgian Grand Prix"	"2013"
"Sebastian"	"Vettel"	"23"	"Italian Grand Prix"	"2013"
"Sebastian"	"Vettel"	"44"	"Singapore Grand Prix"	"2013"
"Sebastian"	"Vettel"	"17"	"Singapore Grand Prix"	"2013"
"Sebastian"	"Vettel"	"11"	"Korean Grand Prix"	"2013"
"Sebastian"	"Vettel"	"31"	"Korean Grand Prix"	"2013"
"Sebastian"	"Vettel"	"14"	"Japanese Grand Prix"	"2013"
"Sebastian"	"Vettel"	"37"	"Japanese Grand Prix"	"2013"
"Sebastian"	"Vettel"	"2"	"Indian Grand Prix"	"2013"
"Sebastian"	"Vettel"	"31"	"Indian Grand Prix"	"2013"
"Sebastian"	"Vettel"	"14"	"Abu Dhabi Grand Prix"	"2013"
"Sebastian"	"Vettel"	"37"	"Abu Dhabi Grand Prix"	"2013"
"Sebastian"	"Vettel"	"27"	"United States Grand Prix"	"2013"
"Sebastian"	"Vettel"	"24"	"Brazilian Grand Prix"	"2013"
"Sebastian"	"Vettel"	"47"	"Brazilian Grand Prix"	"2013"

"Nico"	"Rosberg"	"12"	"Australian Grand Prix"	"2014"
"Nico"	"Rosberg"	"38"	"Australian Grand Prix"	"2014"
"Lewis"	"Hamilton"	"15"	"Malaysian Grand Prix"	"2014"
"Lewis"	"Hamilton"	"33"	"Malaysian Grand Prix"	"2014"
"Lewis"	"Hamilton"	"51"	"Malaysian Grand Prix"	"2014"
"Lewis"	"Hamilton"	"19"	"Bahrain Grand Prix"	"2014"
"Lewis"	"Hamilton"	"41"	"Bahrain Grand Prix"	"2014"
"Lewis"	"Hamilton"	"17"	"Chinese Grand Prix"	"2014"
"Lewis"	"Hamilton"	"38"	"Chinese Grand Prix"	"2014"
"Lewis"	"Hamilton"	"18"	"Spanish Grand Prix"	"2014"
"Lewis"	"Hamilton"	"43"	"Spanish Grand Prix"	"2014"
"Nico"	"Rosberg"	"26"	"Monaco Grand Prix"	"2014"
"Daniel"	"Ricciardo"	"13"	"Canadian Grand Prix"	"2014"
"Daniel"	"Ricciardo"	"37"	"Canadian Grand Prix"	"2014"
"Nico"	"Rosberg"	"11"	"Austrian Grand Prix"	"2014"
"Nico"	"Rosberg"	"40"	"Austrian Grand Prix"	"2014"
"Lewis"	"Hamilton"	"24"	"British Grand Prix"	"2014"
"Lewis"	"Hamilton"	"41"	"British Grand Prix"	"2014"
"Nico"	"Rosberg"	"15"	"German Grand Prix"	"2014"
"Nico"	"Rosberg"	"41"	"German Grand Prix"	"2014"
"Daniel"	"Ricciardo"	"8"	"Hungarian Grand Prix"	"2014"
"Daniel"	"Ricciardo"	"23"	"Hungarian Grand Prix"	"2014"
"Daniel"	"Ricciardo"	"54"	"Hungarian Grand Prix"	"2014"
"Daniel"	"Ricciardo"	"11"	"Belgian Grand Prix"	"2014"
"Daniel"	"Ricciardo"	"27"	"Belgian Grand Prix"	"2014"
"Lewis"	"Hamilton"	"25"	"Italian Grand Prix"	"2014"
"Lewis"	"Hamilton"	"26"	"Singapore Grand Prix"	"2014"
"Lewis"	"Hamilton"	"52"	"Singapore Grand Prix"	"2014"
"Lewis"	"Hamilton"	"13"	"Singapore Grand Prix"	"2014"
"Lewis"	"Hamilton"	"2"	"Japanese Grand Prix"	"2014"
"Lewis"	"Hamilton"	"14"	"Japanese Grand Prix"	"2014"
"Lewis"	"Hamilton"	"35"	"Japanese Grand Prix"	"2014"
"Lewis"	"Hamilton"	"27"	"Russian Grand Prix"	"2014"
"Lewis"	"Hamilton"	"16"	"United States Grand Prix"	"2014"
"Lewis"	"Hamilton"	"33"	"United States Grand Prix"	"2014"
"Nico"	"Rosberg"	"7"	"Brazilian Grand Prix"	"2014"
"Nico"	"Rosberg"	"26"	"Brazilian Grand Prix"	"2014"
"Nico"	"Rosberg"	"50"	"Brazilian Grand Prix"	"2014"
"Lewis"	"Hamilton"	"10"	"Abu Dhabi Grand Prix"	"2014"
"Lewis"	"Hamilton"	"31"	"Abu Dhabi Grand Prix"	"2014"
"Lewis"	"Hamilton"	"25"	"Australian Grand Prix"	"2015"
"Sebastian"	"Vettel"	"17"	"Malaysian Grand Prix"	"2015"
"Sebastian"	"Vettel"	"37"	"Malaysian Grand Prix"	"2015"
"Lewis"	"Hamilton"	"14"	"Chinese Grand Prix"	"2015"
"Lewis"	"Hamilton"	"33"	"Chinese Grand Prix"	"2015"
"Lewis"	"Hamilton"	"15"	"Bahrain Grand Prix"	"2015"
"Lewis"	"Hamilton"	"33"	"Bahrain Grand Prix"	"2015"

"Nico"	"Rosberg"	"15"	"Spanish Grand Prix"	"2015"
"Nico"	"Rosberg"	"45"	"Spanish Grand Prix"	"2015"
"Nico"	"Rosberg"	"37"	"Monaco Grand Prix"	"2015"
"Lewis"	"Hamilton"	"29"	"Canadian Grand Prix"	"2015"
"Nico"	"Rosberg"	"33"	"Austrian Grand Prix"	"2015"
"Lewis"	"Hamilton"	"19"	"British Grand Prix"	"2015"
"Lewis"	"Hamilton"	"43"	"British Grand Prix"	"2015"
"Sebastian"	"Vettel"	"21"	"Hungarian Grand Prix"	"2015"
"Sebastian"	"Vettel"	"43"	"Hungarian Grand Prix"	"2015"
"Sebastian"	"Vettel"	"45"	"Hungarian Grand Prix"	"2015"
"Sebastian"	"Vettel"	"46"	"Hungarian Grand Prix"	"2015"
"Lewis"	"Hamilton"	"13"	"Belgian Grand Prix"	"2015"
"Lewis"	"Hamilton"	"30"	"Belgian Grand Prix"	"2015"
"Lewis"	"Hamilton"	"26"	"Italian Grand Prix"	"2015"
"Sebastian"	"Vettel"	"13"	"Singapore Grand Prix"	"2015"
"Sebastian"	"Vettel"	"37"	"Singapore Grand Prix"	"2015"
"Lewis"	"Hamilton"	"16"	"Japanese Grand Prix"	"2015"
"Lewis"	"Hamilton"	"31"	"Japanese Grand Prix"	"2015"
"Lewis"	"Hamilton"	"32"	"Russian Grand Prix"	"2015"
"Lewis"	"Hamilton"	"18"	"United States Grand Prix"	"2015"
"Lewis"	"Hamilton"	"43"	"United States Grand Prix"	"2015"
"Nico"	"Rosberg"	"26"	"Mexican Grand Prix"	"2015"
"Nico"	"Rosberg"	"46"	"Mexican Grand Prix"	"2015"
"Nico"	"Rosberg"	"13"	"Brazilian Grand Prix"	"2015"
"Nico"	"Rosberg"	"33"	"Brazilian Grand Prix"	"2015"
"Nico"	"Rosberg"	"48"	"Brazilian Grand Prix"	"2015"
"Nico"	"Rosberg"	"10"	"Abu Dhabi Grand Prix"	"2015"
"Nico"	"Rosberg"	"31"	"Abu Dhabi Grand Prix"	"2015"
"Nico"	"Rosberg"	"12"	"Australian Grand Prix"	"2016"
"Nico"	"Rosberg"	"18"	"Australian Grand Prix"	"2016"
"Nico"	"Rosberg"	"13"	"Bahrain Grand Prix"	"2016"
"Nico"	"Rosberg"	"30"	"Bahrain Grand Prix"	"2016"
"Nico"	"Rosberg"	"39"	"Bahrain Grand Prix"	"2016"
"Nico"	"Rosberg"	"20"	"Chinese Grand Prix"	"2016"
"Nico"	"Rosberg"	"36"	"Chinese Grand Prix"	"2016"
"Nico"	"Rosberg"	"21"	"Russian Grand Prix"	"2016"
"Max"	"Verstappen"	"12"	"Spanish Grand Prix"	"2016"
"Max"	"Verstappen"	"34"	"Spanish Grand Prix"	"2016"
"Lewis"	"Hamilton"	"31"	"Monaco Grand Prix"	"2016"
"Lewis"	"Hamilton"	"24"	"Canadian Grand Prix"	"2016"
"Nico"	"Rosberg"	"21"	"European Grand Prix"	"2016"
"Lewis"	"Hamilton"	"21"	"Austrian Grand Prix"	"2016"
"Lewis"	"Hamilton"	"28"	"Austrian Grand Prix"	"2016"
"Lewis"	"Hamilton"	"29"	"Austrian Grand Prix"	"2016"
"Lewis"	"Hamilton"	"54"	"Austrian Grand Prix"	"2016"
"Lewis"	"Hamilton"	"7"	"British Grand Prix"	"2016"
"Lewis"	"Hamilton"	"17"	"British Grand Prix"	"2016"

"Lewis"	"Hamilton"	"16"	"Hungarian Grand Prix"	"2016"
"Lewis"	"Hamilton"	"41"	"Hungarian Grand Prix"	"2016"
"Lewis"	"Hamilton"	"14"	"German Grand Prix"	"2016"
"Lewis"	"Hamilton"	"34"	"German Grand Prix"	"2016"
"Lewis"	"Hamilton"	"47"	"German Grand Prix"	"2016"
"Nico"	"Rosberg"	"9"	"Belgian Grand Prix"	"2016"
"Nico"	"Rosberg"	"26"	"Belgian Grand Prix"	"2016"
"Nico"	"Rosberg"	"24"	"Italian Grand Prix"	"2016"
"Nico"	"Rosberg"	"1"	"Singapore Grand Prix"	"2016"
"Nico"	"Rosberg"	"16"	"Singapore Grand Prix"	"2016"
"Nico"	"Rosberg"	"33"	"Singapore Grand Prix"	"2016"
"Daniel"	"Ricciardo"	"21"	"Malaysian Grand Prix"	"2016"
"Daniel"	"Ricciardo"	"41"	"Malaysian Grand Prix"	"2016"
"Nico"	"Rosberg"	"12"	"Japanese Grand Prix"	"2016"
"Nico"	"Rosberg"	"29"	"Japanese Grand Prix"	"2016"
"Lewis"	"Hamilton"	"11"	"United States Grand Prix"	"2016"
"Lewis"	"Hamilton"	"31"	"United States Grand Prix"	"2016"
"Lewis"	"Hamilton"	"17"	"Mexican Grand Prix"	"2016"
"Lewis"	"Hamilton"	"20"	"Brazilian Grand Prix"	"2016"
"Lewis"	"Hamilton"	"28"	"Brazilian Grand Prix"	"2016"
"Lewis"	"Hamilton"	"7"	"Abu Dhabi Grand Prix"	"2016"
"Lewis"	"Hamilton"	"28"	"Abu Dhabi Grand Prix"	"2016"
"Sebastian"	"Vettel"	"23"	"Australian Grand Prix"	"2017"
"Lewis"	"Hamilton"	"4"	"Chinese Grand Prix"	"2017"
"Lewis"	"Hamilton"	"5"	"Chinese Grand Prix"	"2017"
"Lewis"	"Hamilton"	"6"	"Chinese Grand Prix"	"2017"
"Lewis"	"Hamilton"	"36"	"Chinese Grand Prix"	"2017"
"Sebastian"	"Vettel"	"10"	"Bahrain Grand Prix"	"2017"
"Sebastian"	"Vettel"	"33"	"Bahrain Grand Prix"	"2017"
"Valtteri"	"Bottas"	"27"	"Russian Grand Prix"	"2017"
"Lewis"	"Hamilton"	"21"	"Spanish Grand Prix"	"2017"
"Lewis"	"Hamilton"	"36"	"Spanish Grand Prix"	"2017"
"Sebastian"	"Vettel"	"39"	"Monaco Grand Prix"	"2017"
"Lewis"	"Hamilton"	"32"	"Canadian Grand Prix"	"2017"
"Daniel"	"Ricciardo"	"5"	"Azerbaijan Grand Prix"	"2017"
"Daniel"	"Ricciardo"	"13"	"Azerbaijan Grand Prix"	"2017"
"Daniel"	"Ricciardo"	"17"	"Azerbaijan Grand Prix"	"2017"
"Daniel"	"Ricciardo"	"22"	"Azerbaijan Grand Prix"	"2017"
"Valtteri"	"Bottas"	"41"	"Austrian Grand Prix"	"2017"
"Lewis"	"Hamilton"	"25"	"British Grand Prix"	"2017"
"Sebastian"	"Vettel"	"32"	"Hungarian Grand Prix"	"2017"
"Lewis"	"Hamilton"	"12"	"Belgian Grand Prix"	"2017"
"Lewis"	"Hamilton"	"30"	"Belgian Grand Prix"	"2017"
"Lewis"	"Hamilton"	"32"	"Italian Grand Prix"	"2017"
"Lewis"	"Hamilton"	"1"	"Singapore Grand Prix"	"2017"
"Lewis"	"Hamilton"	"2"	"Singapore Grand Prix"	"2017"
"Lewis"	"Hamilton"	"3"	"Singapore Grand Prix"	"2017"

"Lewis"	"Hamilton"	"29"	"Singapore Grand Prix"	"2017"
"Max"	"Verstappen"	"27"	"Malaysian Grand Prix"	"2017"
"Lewis"	"Hamilton"	"22"	"Japanese Grand Prix"	"2017"
"Lewis"	"Hamilton"	"19"	"United States Grand Prix"	"2017"
"Max"	"Verstappen"	"32"	"Mexican Grand Prix"	"2017"
"Sebastian"	"Vettel"	"1"	"Brazilian Grand Prix"	"2017"
"Sebastian"	"Vettel"	"2"	"Brazilian Grand Prix"	"2017"
"Sebastian"	"Vettel"	"3"	"Brazilian Grand Prix"	"2017"
"Sebastian"	"Vettel"	"28"	"Brazilian Grand Prix"	"2017"
"Valtteri"	"Bottas"	"21"	"Abu Dhabi Grand Prix"	"2017"
"Sebastian"	"Vettel"	"26"	"Australian Grand Prix"	"2018"
"Sebastian"	"Vettel"	"18"	"Bahrain Grand Prix"	"2018"
"Daniel"	"Ricciardo"	"17"	"Chinese Grand Prix"	"2018"
"Daniel"	"Ricciardo"	"31"	"Chinese Grand Prix"	"2018"
"Lewis"	"Hamilton"	"22"	"Azerbaijan Grand Prix"	"2018"
"Lewis"	"Hamilton"	"40"	"Azerbaijan Grand Prix"	"2018"
"Lewis"	"Hamilton"	"25"	"Spanish Grand Prix"	"2018"
"Daniel"	"Ricciardo"	"17"	"Monaco Grand Prix"	"2018"
"Sebastian"	"Vettel"	"37"	"Canadian Grand Prix"	"2018"
"Lewis"	"Hamilton"	"33"	"French Grand Prix"	"2018"
"Max"	"Verstappen"	"15"	"Austrian Grand Prix"	"2018"
"Sebastian"	"Vettel"	"20"	"British Grand Prix"	"2018"
"Sebastian"	"Vettel"	"33"	"British Grand Prix"	"2018"
"Lewis"	"Hamilton"	"42"	"German Grand Prix"	"2018"
"Lewis"	"Hamilton"	"25"	"Hungarian Grand Prix"	"2018"
"Sebastian"	"Vettel"	"1"	"Belgian Grand Prix"	"2018"
"Sebastian"	"Vettel"	"22"	"Belgian Grand Prix"	"2018"
"Lewis"	"Hamilton"	"28"	"Italian Grand Prix"	"2018"
"Lewis"	"Hamilton"	"15"	"Singapore Grand Prix"	"2018"
"Lewis"	"Hamilton"	"14"	"Russian Grand Prix"	"2018"
"Lewis"	"Hamilton"	"24"	"Japanese Grand Prix"	"2018"
"Kimi"	"Räikkönen"	"21"	"United States Grand Prix"	"2018"
"Max"	"Verstappen"	"13"	"Mexican Grand Prix"	"2018"
"Max"	"Verstappen"	"48"	"Mexican Grand Prix"	"2018"
"Lewis"	"Hamilton"	"19"	"Brazilian Grand Prix"	"2018"
"Lewis"	"Hamilton"	"7"	"Abu Dhabi Grand Prix"	"2018"
"Valtteri"	"Bottas"	"23"	"Australian Grand Prix"	"2019"
"Lewis"	"Hamilton"	"13"	"Bahrain Grand Prix"	"2019"
"Lewis"	"Hamilton"	"34"	"Bahrain Grand Prix"	"2019"
"Lewis"	"Hamilton"	"22"	"Chinese Grand Prix"	"2019"
"Lewis"	"Hamilton"	"36"	"Chinese Grand Prix"	"2019"
"Valtteri"	"Bottas"	"12"	"Azerbaijan Grand Prix"	"2019"
"Lewis"	"Hamilton"	"27"	"Spanish Grand Prix"	"2019"
"Lewis"	"Hamilton"	"46"	"Spanish Grand Prix"	"2019"
"Lewis"	"Hamilton"	"11"	"Monaco Grand Prix"	"2019"
"Lewis"	"Hamilton"	"28"	"Canadian Grand Prix"	"2019"
"Lewis"	"Hamilton"	"24"	"French Grand Prix"	"2019"

```
| "Max"      | "Verstappen" | "31"   | "Austrian Grand Prix"      | "2019" |
| "Lewis"    | "Hamilton"   | "20"   | "British Grand Prix"       | "2019" |
| "Max"      | "Verstappen" | "3"    | "German Grand Prix"        | "2019" |
| "Max"      | "Verstappen" | "25"   | "German Grand Prix"        | "2019" |
| "Max"      | "Verstappen" | "29"   | "German Grand Prix"        | "2019" |
| "Max"      | "Verstappen" | "41"   | "German Grand Prix"        | "2019" |
| "Max"      | "Verstappen" | "46"   | "German Grand Prix"        | "2019" |
| "Lewis"    | "Hamilton"   | "31"   | "Hungarian Grand Prix"     | "2019" |
| "Lewis"    | "Hamilton"   | "48"   | "Hungarian Grand Prix"     | "2019" |
| "Charles"  | "Leclerc"    | "21"   | "Belgian Grand Prix"       | "2019" |
| "Charles"  | "Leclerc"    | "20"   | "Italian Grand Prix"       | "2019" |
| "Sebastian" | "Vettel"    | "19"   | "Singapore Grand Prix"     | "2019" |
| "Lewis"    | "Hamilton"   | "28"   | "Russian Grand Prix"       | "2019" |
| "Valtteri" | "Bottas"     | "17"   | "Japanese Grand Prix"      | "2019" |
| "Valtteri" | "Bottas"     | "36"   | "Japanese Grand Prix"      | "2019" |
| "Lewis"    | "Hamilton"   | "23"   | "Mexican Grand Prix"       | "2019" |
| "Valtteri" | "Bottas"     | "14"   | "United States Grand Prix" | "2019" |
| "Valtteri" | "Bottas"     | "35"   | "United States Grand Prix" | "2019" |
| "Max"      | "Verstappen" | "21"   | "Brazilian Grand Prix"     | "2019" |
| "Max"      | "Verstappen" | "44"   | "Brazilian Grand Prix"     | "2019" |
| "Max"      | "Verstappen" | "54"   | "Brazilian Grand Prix"     | "2019" |
| "Lewis"    | "Hamilton"   | "26"   | "Abu Dhabi Grand Prix"     | "2019" |
+-----------------------------------------------------------------------------+
```

Appendix J: Output of Query to Return Date of Birth and Average Race Position of Drivers

```
+-------------------------------------------------------------------------------+
| forename      | surname              | dateOfBirth | racePosition      |
+-------------------------------------------------------------------------------+
| "Philippe"    | "Étancelin"          | "1896-12-28" | 7.333333333333333 |
| "Luigi"       | "Fagioli"            | "1898-06-09" | 3.2857142857142856 |
| "Clemente"    | "Biondetti"          | "1898-08-18" | NULL              |
| "Arthur"      | "Legat"              | "1898-11-01" | 13.0              |
| "Louis"       | "Chiron"             | "1899-08-03" | 8.0               |
| "Piero"       | "Dusio"              | "1899-10-13" | NULL              |
| "Adolf"       | "Brudes"             | "1899-10-15" | NULL              |
| "Bill"        | "Aston"              | "1900-03-29" | NULL              |
| "Hans"        | "von Stuck"          | "1900-12-27" | 14.0              |
| "Rudolf"      | "Schoeller"          | "1902-04-27" | NULL              |
| "Chet"        | "Miller"             | "1902-07-19" | NULL              |
| "Ernst"       | "Klodwig"            | "1903-05-23" | 13.5              |
| "Felice"      | "Bonetto"            | "1903-06-09" | 4.333333333333333 |
| "Juan"        | "Jover"              | "1903-11-23" | NULL              |
| "Yves"        | "Cabantous"          | "1904-10-08" | 8.874999999999998 |
| "Georges"     | "Grignard"           | "1905-07-25" | NULL              |
| "Louis"       | "Rosier"             | "1905-11-05" | 7.499999999999999 |
| "Pierre"      | "Levegh"             | "1905-12-22" | 8.0               |
| "Toni"        | "Ulmen"              | "1906-01-25" | 8.0               |
| "Mauri"       | "Rose"               | "1906-05-26" | 3.0               |
| "Cuth"        | "Harrison"           | "1906-07-06" | 7.0               |
| "Franco"      | "Comotti"            | "1906-07-24" | 12.0              |
| "George"      | "Connor"             | "1906-08-16" | 8.0               |
| "Raymond"     | "Sommer"             | "1906-08-31" | 4.0               |
| "Eitel"       | "Cantoni"            | "1906-10-04" | 11.0              |
| "Piero"       | "Taruffi"            | "1906-10-12" | 4.083333333333333 |
| "Nino"        | "Farina"             | "1906-10-30" | 2.8148148148148144 |
| "Rudolf"      | "Krause"             | "1907-03-30" | 14.0              |
| "Eugène"      | "Chaboud"            | "1907-04-12" | 6.5               |
| "Ernst"       | "Loof"               | "1907-07-04" | NULL              |
| "Chico"       | "Landi"              | "1907-07-14" | 7.0               |
| "Roger"       | "Loyer"              | "1907-08-05" | NULL              |
| "Philip"      | "Fotheringham-Parker" | "1907-09-22" | NULL              |
| "Bill"        | "Holland"            | "1907-12-18" | 8.5               |
| "Bill"        | "Cantrell"           | "1908-01-31" | NULL              |
| "Albert"      | "Scherrer"           | "1908-02-28" | 9.0               |
| "Robert"      | "O'Brien"            | "1908-04-11" | 14.0              |
| "Franco"      | "Rol"                | "1908-06-05" | 9.0               |
| "Marcel"      | "Balsa"              | "1909-01-01" | NULL              |
| "Bill"        | "Schindler"          | "1909-03-06" | 14.0              |
```

"Bill"	"Whitehouse"	"1909-04-01"	NULL
"Hermann"	"Lang"	"1909-04-06"	5.0
"Luigi"	"Villoresi"	"1909-05-16"	4.277777777777778
"Dorino"	"Serafini"	"1909-07-22"	2.0
"Charles"	"Pozzi"	"1909-08-27"	6.0
"Piero"	"Scotti"	"1909-11-11"	NULL
"David"	"Murray"	"1909-12-28"	NULL
"Henri"	"Louveau"	"1910-01-25"	NULL
"Peter"	"Hirt"	"1910-03-30"	9.0
"Kurt"	"Kuhnke"	"1910-04-30"	NULL
"Jimmy"	"Jackson"	"1910-07-25"	15.0
"Guy"	"Mairesse"	"1910-08-10"	11.5
"Lee"	"Wallard"	"1910-09-07"	3.5
"Karl"	"Kling"	"1910-09-16"	4.166666666666667
"Travis"	"Webb"	"1910-10-08"	20.0
"Carl"	"Forberg"	"1911-03-04"	7.0
"Consalvo"	"Sanesi"	"1911-03-28"	6.666666666666667
"Paul"	"Pietsch"	"1911-06-20"	NULL
"Juan"	"Fangio"	"1911-06-24"	2.2500000000000004
"Reg"	"Parnell"	"1911-07-02"	4.75
"Len"	"Duncan"	"1911-07-25"	NULL
"Ken"	"Richardson"	"1911-08-21"	NULL
"Nello"	"Pagani"	"1911-10-11"	7.0
"Walt"	"Brown"	"1911-12-30"	19.0
"Leslie"	"Johnson"	"1912-03-22"	NULL
"Joie"	"Chitwood"	"1912-04-14"	5.0
"Rudi"	"Fischer"	"1912-04-19"	7.666666666666667
"Myron"	"Fohr"	"1912-06-17"	11.0
"Erwin"	"Bauer"	"1912-07-17"	NULL
"Peter"	"Walker"	"1912-10-07"	7.0
"Alfredo"	"Pián"	"1912-10-21"	NULL
"Roger"	"Laurent"	"1913-02-21"	9.0
"Duke"	"Nalon"	"1913-03-02"	11.0
"Joe"	"Kelly"	"1913-03-13"	NULL
"Jack"	"Fairman"	"1913-03-15"	5.666666666666667
"George"	"Abecassis"	"1913-03-21"	NULL
"Duke"	"Dinsmore"	"1913-04-10"	16.5
"Duane"	"Carter"	"1913-05-05"	8.444444444444445
"Theo"	"Helfrich"	"1913-05-13"	12.0
"Henry"	"Banks"	"1913-06-14"	12.5
"Oscar"	"Gálvez"	"1913-08-17"	5.0
"Fred"	"Agabashian"	"1913-08-21"	7.333333333333333
"Luigi"	"Piotti"	"1913-10-27"	8.0
"Walt"	"Ader"	"1913-12-15"	22.0
"Bob"	"Gerard"	"1914-01-19"	8.714285714285715
"Bayliss"	"Levrett"	"1914-02-14"	NULL
"Élie"	"Bayol"	"1914-02-28"	5.5

"Paul"	"Russo"	"1914-04-10"	6.0
"Cal"	"Niday"	"1914-04-29"	10.0
"John"	"James"	"1914-05-10"	NULL
"Toulo"	"de Graffenried"	"1914-05-18"	6.999999999999999
"Carl"	"Scarborough"	"1914-07-03"	12.0
"Sam"	"Hanks"	"1914-07-13"	4.0
"Prince"	"Bira"	"1914-07-15"	7.4
"Carlos"	"Menditeguy"	"1914-08-10"	4.75
"Josef"	"Peters"	"1914-09-16"	NULL
"Peter"	"Whitehead"	"1914-11-12"	7.6
"Brian"	"Shawe Taylor"	"1915-01-28"	9.0
"Johnny"	"McDowell"	"1915-01-29"	19.5
"Arthur"	"Owen"	"1915-03-23"	NULL
"Eugène"	"Martin"	"1915-03-24"	NULL
"Robin"	"Montgomerie-Charrington"	"1915-06-23"	NULL
"Mack"	"Hellings"	"1915-09-14"	13.0
"Ray"	"Crawford"	"1915-10-26"	NULL
"Joe"	"Fry"	"1915-10-26"	10.0
"Helmut"	"Niedermayr"	"1915-11-29"	9.0
"Cliff"	"Griffith"	"1916-02-06"	9.5
"Ken"	"Wharton"	"1916-03-21"	7.375
"Phil"	"Cade"	"1916-06-12"	NULL
"Gene"	"Force"	"1916-06-15"	28.0
"Leslie"	"Thorne"	"1916-06-23"	14.0
"Gino"	"Bianco"	"1916-07-22"	18.0
"Johnny"	"Claes"	"1916-08-11"	10.384615384615385
"Dennis"	"Poore"	"1916-08-19"	8.0
"Tony"	"Bettenhausen"	"1916-09-12"	9.625
"Toni"	"Branca"	"1916-09-15"	10.5
"Paul"	"Emery"	"1916-11-12"	NULL
"Edgar"	"Barth"	"1917-01-26"	8.333333333333334
"Paul"	"Frère"	"1917-01-30"	5.8
"Danny"	"Kladis"	"1917-02-10"	NULL
"Robert"	"La Caze"	"1917-02-26"	14.0
"Oswald"	"Karch"	"1917-03-06"	NULL
"Robert"	"Manzon"	"1917-04-12"	6.909090909090908
"Jean"	"Lucas"	"1917-04-25"	NULL
"George"	"Fonder"	"1917-06-22"	17.0
"John"	"Fitch"	"1917-08-04"	9.0
"Johnnie"	"Tolan"	"1917-10-22"	13.0
"Maurice"	"Trintignant"	"1917-10-30"	6.136363636363638
"Ken"	"Downing"	"1917-12-05"	9.0
"Ottorino"	"Volonterio"	"1917-12-07"	11.0
"David"	"Hampshire"	"1917-12-29"	9.0
"Art"	"Cross"	"1918-01-24"	6.0
"George"	"Constantine"	"1918-02-22"	NULL
"Dick"	"Gibson"	"1918-04-16"	NULL

"Johnnie"	"Parsons"	"1918-07-04"	9.0
"Fred"	"Wacker"	"1918-07-10"	7.5
"Alberto"	"Ascari"	"1918-07-13"	2.181818181818182
"Bill"	"Homeier"	"1918-08-31"	9.0
"Georges"	"Berger"	"1918-09-14"	NULL
"Johnny"	"Mantz"	"1918-09-18"	17.0
"Horace"	"Gould"	"1918-09-20"	9.4
"André"	"Pilette"	"1918-10-06"	7.714285714285714
"Max"	"de Terra"	"1918-10-06"	8.0
"Frank"	"Armi"	"1918-10-12"	19.0
"Tony"	"Rolt"	"1918-10-16"	NULL
"Ken"	"Miles"	"1918-11-01"	NULL
"Bill"	"Vukovich"	"1918-12-13"	6.333333333333333
"Eddie"	"Johnson"	"1919-02-10"	10.571428571428571
"André"	"Guelfi"	"1919-05-06"	15.0
"Shorty"	"Templeman"	"1919-08-12"	17.0
"Dries"	"van der Lof"	"1919-08-23"	NULL
"Lance"	"Macklin"	"1919-09-02"	10.2
"Roberto"	"Bonomi"	"1919-09-30"	11.0
"Cecil"	"Green"	"1919-09-30"	4.0
"Jan"	"Flinterman"	"1919-10-02"	9.0
"Jack"	"McGrath"	"1919-10-08"	7.2
"Chuck"	"Stevenson"	"1919-10-15"	13.5
"Walt"	"Hansgen"	"1919-10-28"	5.0
"Hans"	"Klenk"	"1919-10-28"	11.0
"Eric"	"Thompson"	"1919-11-04"	5.0
"Alan"	"Brown"	"1919-11-20"	11.5
"Bob"	"Drake"	"1919-12-14"	13.0
"André"	"Simon"	"1920-01-05"	8.0
"Alberto"	"Crespo"	"1920-01-16"	NULL
"Tony"	"Gaze"	"1920-02-03"	15.0
"Mike"	"Magill"	"1920-02-08"	NULL
"Jack"	"Turner"	"1920-02-12"	11.0
"Walt"	"Faulkner"	"1920-02-16"	10.25
"Tony"	"Crook"	"1920-02-16"	21.0
"Alfonso"	"Thiele"	"1920-04-05"	NULL
"Al"	"Keller"	"1920-04-11"	12.5
"Duncan"	"Hamilton"	"1920-04-30"	9.5
"Syd"	"van der Vyver"	"1920-06-01"	NULL
"Don"	"Branson"	"1920-06-02"	4.0
"Keith"	"Andrews"	"1920-06-15"	NULL
"Eric"	"Brandon"	"1920-07-18"	12.5
"Clemar"	"Bucci"	"1920-09-04"	NULL
"Jackie"	"Holmes"	"1920-09-04"	23.0
"Kenneth"	"McAlpine"	"1920-09-21"	14.5
"Frank"	"Dochnal"	"1920-10-08"	NULL
"Doug"	"Serrurier"	"1920-12-09"	11.0

"Les"	"Leston"	"1920-12-16"	NULL
"Rodger"	"Ward"	"1921-01-10"	6.75
"Jean"	"Behra"	"1921-02-16"	4.919999999999999
"Ernie"	"McCoy"	"1921-02-19"	12.0
"Marshall"	"Teague"	"1921-02-22"	11.0
"Piero"	"Carini"	"1921-03-06"	NULL
"Paco"	"Godia"	"1921-03-21"	7.000000000000001
"François"	"Picard"	"1921-04-26"	NULL
"Geoff"	"Crossley"	"1921-05-11"	9.0
"Aldo"	"Gordini"	"1921-05-20"	NULL
"Ettore"	"Chimeri"	"1921-06-04"	NULL
"Dennis"	"Taylor"	"1921-06-12"	NULL
"Harry"	"Schell"	"1921-06-29"	6.741935483870969
"Sam"	"Tingle"	"1921-08-24"	10.5
"Ian"	"Raby"	"1921-09-22"	11.0
"Giorgio"	"Scarlatti"	"1921-10-02"	8.2
"Mike"	"Nazaruk"	"1921-10-02"	3.5
"Al"	"Pease"	"1921-10-15"	NULL
"Manny"	"Ayulo"	"1921-10-20"	12.25
"Kurt"	"Adolff"	"1921-11-05"	NULL
"Günther"	"Bechem"	"1921-12-21"	NULL
"Alain"	"de Changy"	"1922-02-05"	NULL
"Willi"	"Heeks"	"1922-02-13"	NULL
"Jesús"	"Iglesias"	"1922-02-22"	NULL
"Andy"	"Linden"	"1922-04-05"	8.4
"Johnny"	"Thomson"	"1922-04-09"	6.0
"Graham"	"Whitehead"	"1922-04-15"	12.0
"Roy"	"Salvadori"	"1922-05-12"	6.789473684210527
"Jacques"	"Pollet"	"1922-07-02"	8.5
"Fritz"	"Riess"	"1922-07-11"	7.0
"Leslie"	"Marr"	"1922-08-14"	13.0
"José Froilán"	"González"	"1922-10-05"	3.0
"Azdrubal"	"Fontes"	"1922-12-26"	NULL
"Jean"	"Lucienbonnet"	"1923-01-07"	NULL
"Carroll"	"Shelby"	"1923-01-11"	7.75
"Jud"	"Larson"	"1923-01-21"	8.0
"Theo"	"Fitzau"	"1923-02-10"	NULL
"Giulio"	"Cabianca"	"1923-02-19"	9.5
"Brian"	"Naylor"	"1923-03-24"	13.0
"Geoff"	"Duke"	"1923-03-29"	NULL
"Chuck"	"Weyant"	"1923-04-03"	13.0
"Ernesto"	"Prinoth"	"1923-04-15"	NULL
"Jorge"	"Daponte"	"1923-06-05"	11.0
"Ivor"	"Bueb"	"1923-06-06"	10.666666666666666
"Jim"	"Rigsby"	"1923-06-06"	12.0
"Ron"	"Flockhart"	"1923-06-16"	7.0
"Adolfo"	"Cruz"	"1923-06-28"	NULL

"Óscar"	"González"	"1923-11-10"	6.0
"Chuck"	"Daigh"	"1923-11-29"	10.0
"Ken"	"Kavanagh"	"1923-12-12"	NULL
"Onofre"	"Marimón"	"1923-12-19"	5.0
"Dick"	"Rathmann"	"1924-01-06"	18.0
"Pablo"	"Birger"	"1924-01-07"	NULL
"Olivier"	"Gendebien"	"1924-01-12"	6.272727272727273
"John"	"Riseley-Prichard"	"1924-01-17"	NULL
"Bob"	"Christie"	"1924-04-04"	12.0
"Alberto"	"Uria"	"1924-07-11"	6.0
"Luigi"	"Musso"	"1924-07-28"	4.214285714285714
"Jimmy"	"Daywalt"	"1924-08-28"	9.666666666666666
"Bobby"	"Grim"	"1924-09-04"	16.0
"George"	"Amick"	"1924-10-24"	2.0
"Antonio"	"Creus"	"1924-10-28"	NULL
"Roberto"	"Mieres"	"1924-12-03"	5.125
"John"	"Love"	"1924-12-07"	8.666666666666666
"Giovanni"	"de Riu"	"1925-03-10"	NULL
"Don"	"Freeland"	"1925-03-25"	11.2
"Joe"	"James"	"1925-05-23"	13.0
"Len"	"Sutton"	"1925-08-09"	30.0
"Bobby"	"Ball"	"1925-08-26"	5.0
"Paul"	"Goldsmith"	"1925-10-02"	4.0
"Eddie"	"Russo"	"1925-11-19"	26.0
"Hernando"	"da Silva Ramos"	"1925-12-07"	7.0
"Jay"	"Chamberlain"	"1925-12-29"	15.0
"Pat"	"Flaherty"	"1926-01-06"	7.0
"Gene"	"Hartley"	"1926-01-28"	13.428571428571429
"Jimmy"	"Bryan"	"1926-01-28"	9.142857142857144
"Jack"	"Brabham"	"1926-04-02"	4.630136986301369
"Pete"	"Lovely"	"1926-04-11"	9.0
"André"	"Testut"	"1926-04-13"	NULL
"Rob"	"Schroeder"	"1926-05-11"	10.0
"Peter"	"Broeker"	"1926-05-15"	7.0
"Bob"	"Sweikert"	"1926-05-20"	7.0
"Chuck"	"Arnold"	"1926-05-30"	15.0
"Wolfgang"	"Seidel"	"1926-07-04"	14.0
"Tony"	"Settember"	"1926-07-10"	9.5
"Piero"	"Drogo"	"1926-08-08"	8.0
"Johnny"	"Boyd"	"1926-08-19"	10.5
"Roberto"	"Lippi"	"1926-10-17"	NULL
"Larry"	"Crockett"	"1926-10-23"	9.0
"Jacques"	"Swaters"	"1926-10-30"	8.333333333333334
"Bob"	"Veith"	"1926-11-01"	9.0
"Art"	"Bisch"	"1926-11-10"	NULL
"Maria"	"de Filippis"	"1926-11-11"	10.0
"Ed"	"Elisian"	"1926-12-09"	18.0

"Dempsey"	"Wilson"	"1927-03-11"	33.0
"Al"	"Herman"	"1927-03-15"	17.333333333333332
"Phil"	"Hill"	"1927-04-20"	5.151515151515152
"Archie"	"Scott Brown"	"1927-05-13"	NULL
"Eddie"	"Sachs"	"1927-05-28"	21.0
"Charles"	"de Tornaco"	"1927-06-07"	7.0
"Bill"	"Cheesbourg"	"1927-06-12"	10.0
"Herbert"	"MacKay-Fraser"	"1927-06-23"	NULL
"Jim"	"McWithey"	"1927-07-04"	22.5
"Heini"	"Walter"	"1927-07-28"	14.0
"John"	"Rhodes"	"1927-08-18"	NULL
"Roberto"	"Bussinello"	"1927-10-04"	13.0
"Bill"	"Mackey"	"1927-12-15"	NULL
"Hap"	"Sharp"	"1928-01-01"	10.25
"Lloyd"	"Ruby"	"1928-01-12"	7.0
"Hans"	"Herrmann"	"1928-02-23"	8.0
"Gino"	"Munaron"	"1928-04-02"	14.0
"Bud"	"Tingelstad"	"1928-04-04"	9.0
"Desmond"	"Titterington"	"1928-05-01"	NULL
"Wolfgang"	"von Trips"	"1928-05-04"	4.736842105263158
"Jo"	"Schlesser"	"1928-05-18"	NULL
"Colin"	"Chapman"	"1928-05-19"	NULL
"Umberto"	"Maglioli"	"1928-06-05"	6.0
"Alessandro"	"de Tomaso"	"1928-07-10"	9.0
"Elmer"	"George"	"1928-07-15"	NULL
"Jim"	"Rathmann"	"1928-07-16"	7.999999999999999
"Don"	"Beauman"	"1928-07-26"	11.0
"Christian"	"Goethals"	"1928-08-04"	NULL
"Gerino"	"Gerini"	"1928-08-10"	8.75
"Willy"	"Mairesse"	"1928-10-01"	4.666666666666667
"Bob"	"Scott"	"1928-10-04"	15.0
"Pat"	"O'Connor"	"1928-10-09"	11.333333333333334
"Alfonso"	"de Portago"	"1928-10-11"	5.666666666666666
"Ted"	"Whiteaway"	"1928-11-01"	NULL
"André"	"Milhoux"	"1928-12-09"	NULL
"Red"	"Amick"	"1929-01-19"	11.0
"Jerry"	"Hoyt"	"1929-01-29"	14.5
"Graham"	"Hill"	"1929-02-15"	6.009803921568626
"Paul"	"England"	"1929-03-28"	NULL
"Mike"	"Hawthorn"	"1929-04-10"	3.696969696969697
"Sergio"	"Mantovani"	"1929-05-22"	6.666666666666666
"Rob"	"Slotemaker"	"1929-06-13"	NULL
"Harry"	"Blanchard"	"1929-06-13"	7.0
"Rodney"	"Nuckey"	"1929-06-26"	11.0
"Ian"	"Stewart"	"1929-07-15"	NULL
"John"	"Barber"	"1929-07-22"	8.0
"Jimmy"	"Davies"	"1929-08-08"	10.25

"Stirling"	"Moss"	"1929-09-17"	3.4594594594594588
"Jimmy"	"Reece"	"1929-11-17"	9.75
"Menato"	"Boffa"	"1930-01-04"	NULL
"Luki"	"Botha"	"1930-01-16"	NULL
"John"	"Campbell-Jones"	"1930-01-21"	12.0
"Heinz"	"Schiller"	"1930-01-25"	NULL
"Jo"	"Bonnier"	"1930-01-31"	7.320000000000002
"Troy"	"Ruttman"	"1930-03-11"	10.0
"Stuart"	"Lewis-Evans"	"1930-04-20"	4.333333333333334
"Innes"	"Ireland"	"1930-06-12"	6.884615384615384
"Mike"	"Sparken"	"1930-06-16"	7.0
"Ian"	"Burgess"	"1930-07-06"	11.11111111111111
"Guy"	"Ligier"	"1930-07-12"	9.666666666666668
"Richie"	"Ginther"	"1930-08-05"	5.282051282051283
"Don"	"Edmunds"	"1930-09-23"	NULL
"Frank"	"Gardner"	"1930-10-01"	10.333333333333334
"Bernard"	"Collomb"	"1930-10-07"	10.0
"Eugenio"	"Castellotti"	"1930-10-10"	5.000000000000001
"Bernie"	"Ecclestone"	"1930-10-28"	NULL
"David"	"Piper"	"1930-12-02"	12.0
"Jimmy"	"Stewart"	"1931-03-06"	NULL
"Dan"	"Gurney"	"1931-04-13"	5.295454545454546
"Vic"	"Wilson"	"1931-04-14"	NULL
"David"	"Clapham"	"1931-05-18"	NULL
"Bruce"	"Halford"	"1931-05-18"	9.5
"Bob"	"Anderson"	"1931-05-19"	8.200000000000001
"Tony"	"Marsh"	"1931-07-20"	12.666666666666666
"Mike"	"Parkes"	"1931-09-24"	3.0
"Peter"	"Collins"	"1931-11-06"	4.1875
"Cliff"	"Allison"	"1932-02-08"	6.699999999999999
"Tony"	"Brooks"	"1932-02-25"	5.045454545454547
"Masten"	"Gregory"	"1932-02-29"	8.0
"Fred"	"Gamble"	"1932-03-17"	10.0
"Ray"	"Reed"	"1932-04-30"	NULL
"Bob"	"Said"	"1932-05-05"	NULL
"Luigi"	"Taramazzo"	"1932-05-05"	NULL
"Tim"	"Parnell"	"1932-06-25"	10.0
"Carlo"	"Abate"	"1932-07-10"	NULL
"Mike"	"MacDowel"	"1932-09-13"	7.0
"Cesare"	"Perdisa"	"1932-10-21"	5.571428571428571
"Alex"	"Soler-Roig"	"1932-10-29"	NULL
"Jerry"	"Unser"	"1932-11-15"	NULL
"Nasif"	"Estéfano"	"1932-11-18"	14.0
"Alex"	"Blignaut"	"1932-11-30"	NULL
"Gaetano"	"Starrabba"	"1932-12-03"	NULL
"Jim"	"Hurtubise"	"1932-12-05"	18.0
"Henry"	"Taylor"	"1932-12-16"	9.285714285714286

"Nino"	"Vaccarella"	"1933-03-04"	12.0
"John"	"Taylor"	"1933-03-23"	9.0
"Renato"	"Pirocchi"	"1933-03-26"	12.0
"Billy"	"Garrett"	"1933-04-24"	16.0
"Bob"	"Bondurant"	"1933-04-27"	7.25
"John"	"Cannon"	"1933-06-21"	14.0
"Chris"	"Lawrence"	"1933-07-27"	11.0
"Colin"	"Davis"	"1933-07-29"	11.0
"Alan"	"Stacey"	"1933-08-29"	8.0
"Bill"	"Moss"	"1933-09-04"	NULL
"Ludovico"	"Scarfiotti"	"1933-10-18"	5.0
"Peter"	"Arundell"	"1933-11-08"	6.000000000000001
"Alberto Rodriguez"	"Larreta"	"1934-01-14"	9.0
"Mário de Araújo"	"Cabral"	"1934-01-15"	10.0
"Giorgio"	"Bassi"	"1934-01-20"	NULL
"George"	"Follmer"	"1934-01-27"	10.571428571428571
"Juan Manuel"	"Bordeu"	"1934-01-28"	NULL
"Ernesto"	"Brambilla"	"1934-01-31"	NULL
"John"	"Surtees"	"1934-02-11"	4.759259259259259
"Bobby"	"Unser"	"1934-02-20"	NULL
"Carel Godin"	"de Beaufort"	"1934-04-10"	10.086956521739133
"Brian"	"Gubby"	"1934-04-17"	NULL
"Mike"	"Taylor"	"1934-04-24"	NULL
"Paddy"	"Driver"	"1934-05-13"	NULL
"Peter"	"Monteverdi"	"1934-06-07"	NULL
"Tom"	"Bridger"	"1934-06-24"	NULL
"Michael"	"May"	"1934-08-18"	11.0
"Peter"	"Ashdown"	"1934-10-16"	12.0
"Lucien"	"Bianchi"	"1934-11-10"	8.666666666666666
"Jackie"	"Pretorius"	"1934-11-22"	NULL
"Wayne"	"Weiler"	"1934-12-09"	24.0
"Giancarlo"	"Baghetti"	"1934-12-25"	8.0
"Anthony"	"Foyt"	"1935-01-16"	17.5
"Peter"	"de Klerk"	"1935-03-16"	10.5
"Hubert"	"Hahne"	"1935-03-28"	10.0
"Vic"	"Elford"	"1935-06-10"	7.0
"Carlo"	"Facetti"	"1935-06-26"	NULL
"John"	"Cordts"	"1935-07-23"	NULL
"Jim"	"Hall"	"1935-07-23"	7.875
"Massimo"	"Natili"	"1935-07-28"	NULL
"Gerhard"	"Mitter"	"1935-08-30"	6.5
"Trevor"	"Blokdyk"	"1935-11-30"	12.0
"Lorenzo"	"Bandini"	"1935-12-21"	6.142857142857142
"Bill"	"Brack"	"1935-12-26"	NULL
"Moisés"	"Solana"	"1935-12-26"	11.0
"Lance"	"Reventlow"	"1936-02-24"	NULL
"Jim"	"Clark"	"1936-03-04"	3.8400000000000007

"Bruce"	"Kessler"	"1936-03-23"	NULL
"Ronnie"	"Bucknum"	"1936-04-05"	9.75
"Denny"	"Hulme"	"1936-06-18"	4.794871794871795
"Jo"	"Siffert"	"1936-07-07"	7.134615384615385
"Gerry"	"Ashmore"	"1936-07-25"	16.0
"Brian"	"Hart"	"1936-09-07"	NULL
"Dave"	"Charlton"	"1936-10-27"	15.0
"Jackie"	"Lewis"	"1936-11-01"	8.0
"Skip"	"Barber"	"1936-11-16"	16.0
"Ben"	"Pon"	"1936-12-09"	NULL
"Trevor"	"Taylor"	"1936-12-26"	8.416666666666668
"Mike"	"Spence"	"1936-12-30"	6.4
"Luiz"	"Bueno"	"1937-01-16"	12.0
"Bruce"	"Johnstone"	"1937-01-30"	9.0
"Tony"	"Shelly"	"1937-02-02"	NULL
"Tony"	"Maggs"	"1937-02-09"	6.666666666666665
"Roger"	"Penske"	"1937-02-20"	8.5
"Brian"	"Redman"	"1937-03-09"	7.714285714285714
"Mark"	"Donohue"	"1937-03-18"	7.375
"Jean-Pierre"	"Beltoise"	"1937-04-26"	6.673469387755102
"Gus"	"Hutchison"	"1937-04-26"	NULL
"Bruce"	"McLaren"	"1937-08-30"	4.6825396825396846
"Gary"	"Hocking"	"1937-09-30"	NULL
"David"	"Prophet"	"1937-10-09"	14.0
"Paul"	"Hawkins"	"1937-10-12"	9.5
"Günther"	"Seiffert"	"1937-10-18"	NULL
"Giacomo"	"Russo"	"1937-10-23"	9.0
"Vittorio"	"Brambilla"	"1937-11-11"	9.405405405405407
"Chris"	"Bristow"	"1937-12-02"	10.0
"Carlo"	"Franchi"	"1938-01-01"	NULL
"Keith"	"Greene"	"1938-01-05"	15.0
"Alan"	"Rees"	"1938-01-12"	8.0
"Timmy"	"Mayer"	"1938-02-22"	NULL
"Fritz"	"d'Orey"	"1938-03-25"	10.0
"Peter"	"Westbury"	"1938-05-26"	NULL
"Eppie"	"Wietzes"	"1938-05-28"	NULL
"Ernie"	"Pieterse"	"1938-07-04"	10.0
"Neville"	"Lederle"	"1938-09-25"	6.0
"Brausch"	"Niemann"	"1939-01-07"	14.0
"Peter"	"Revson"	"1939-02-27"	5.761904761904762
"Basil"	"van Rooyen"	"1939-04-19"	NULL
"Mike"	"Harris"	"1939-05-25"	NULL
"Dieter"	"Quester"	"1939-05-30"	9.0
"David"	"Hobbs"	"1939-06-09"	8.833333333333334
"Jackie"	"Stewart"	"1939-06-11"	2.952380952380953
"Clay"	"Regazzoni"	"1939-09-05"	5.999999999999999
"Chris"	"Craft"	"1939-11-17"	NULL

"Conny"	"Andersson"	"1939-12-28"	NULL
"Pedro"	"Rodríguez"	"1940-01-18"	5.3103448275862055
"Kunimitsu"	"Takahashi"	"1940-01-29"	9.0
"Peter"	"Gethin"	"1940-02-21"	9.09090909090909
"Mario"	"Andretti"	"1940-02-28"	5.709677419354841
"Graham"	"McRae"	"1940-03-05"	NULL
"Mike"	"Hailwood"	"1940-04-02"	7.925925925925926
"Richard"	"Attwood"	"1940-04-04"	8.749999999999998
"Mike"	"Beuttler"	"1940-04-13"	11.46153846153846
"Kurt"	"Ahrens"	"1940-04-19"	12.0
"Gérard"	"Larrousse"	"1940-05-23"	NULL
"Peter"	"Ryan"	"1940-06-10"	9.0
"Nanni"	"Galli"	"1940-10-02"	11.999999999999998
"Lella"	"Lombardi"	"1941-03-26"	12.571428571428571
"Silvio"	"Moser"	"1941-04-24"	7.25
"David"	"Walker"	"1941-06-10"	13.0
"Ernie"	"de Vos"	"1941-07-01"	NULL
"Clive"	"Puzey"	"1941-07-11"	NULL
"Ignazio"	"Giunti"	"1941-08-30"	8.333333333333334
"Reine"	"Wisell"	"1941-09-30"	6.5
"Andrea"	"de Adamich"	"1941-10-03"	9.583333333333332
"John"	"Nicholson"	"1941-10-06"	17.0
"Derek"	"Bell"	"1941-10-31"	8.5
"Howden"	"Ganley"	"1941-12-24"	8.476190476190476
"Johnny"	"Servoz-Gavin"	"1942-01-18"	5.25
"Ricardo"	"Rodríguez"	"1942-02-14"	8.0
"Gijs"	"van Lennep"	"1942-03-16"	9.714285714285714
"Carlos"	"Reutemann"	"1942-04-12"	5.216494845360824
"Jochen"	"Rindt"	"1942-04-18"	4.2
"Danny"	"Ongais"	"1942-05-21"	7.0
"Piers"	"Courage"	"1942-05-27"	5.555555555555555
"Robin"	"Widdows"	"1942-05-27"	NULL
"Jo"	"Vonlanthen"	"1942-05-31"	NULL
"Chris"	"Irwin"	"1942-06-27"	7.0
"Teddy"	"Pilette"	"1942-07-26"	17.0
"Jackie"	"Oliver"	"1942-08-14"	7.5625
"Tom"	"Belsø"	"1942-08-27"	8.0
"Alessandro"	"Pesenti-Rossi"	"1942-08-31"	14.333333333333334
"Henri"	"Pescarolo"	"1942-09-25"	9.838709677419358
"Jean-Pierre"	"Jabouille"	"1942-10-01"	8.700000000000001
"Jonathan"	"Williams"	"1942-10-26"	8.0
"Guy"	"Edwards"	"1942-12-30"	13.125
"Tony"	"Trimmer"	"1943-01-24"	NULL
"François"	"Mazet"	"1943-02-24"	13.0
"Arturo"	"Merzario"	"1943-03-11"	9.187500000000002
"Mike"	"Fisher"	"1943-03-13"	11.0
"Hans"	"Heyer"	"1943-03-16"	NULL

"Vern"	"Schuppan"	"1943-03-19"	12.5
"Tom"	"Jones"	"1943-04-26"	NULL
"Helmut"	"Marko"	"1943-04-27"	11.142857142857142
"Alan"	"Rollinson"	"1943-05-15"	NULL
"John"	"Miles"	"1943-06-14"	7.5
"Rolf"	"Stommelen"	"1943-07-11"	10.323529411764707
"Chris"	"Amon"	"1943-07-20"	6.9019607843137285
"Max"	"Jean"	"1943-07-27"	NULL
"Leo"	"Kinnunen"	"1943-08-05"	NULL
"Tim"	"Schenken"	"1943-09-26"	10.125
"Bertil"	"Roos"	"1943-10-12"	NULL
"Jacques"	"Laffite"	"1943-11-21"	5.9042553191489375
"Wilson"	"Fittipaldi"	"1943-12-25"	10.823529411764708
"Ronnie"	"Peterson"	"1944-02-14"	6.070422535211268
"François"	"Cevert"	"1944-02-25"	5.03448275862069
"Jac"	"Nelleman"	"1944-04-19"	NULL
"José"	"Dolhem"	"1944-04-26"	NULL
"Sam"	"Posey"	"1944-05-26"	12.0
"Emilio"	"Zapico"	"1944-05-27"	NULL
"Dave"	"Morgan"	"1944-08-07"	18.0
"Patrick"	"Depailler"	"1944-08-09"	5.5
"Divina"	"Galica"	"1944-08-13"	NULL
"Eddie"	"Keizan"	"1944-09-12"	13.5
"Richard"	"Robarts"	"1944-09-22"	16.0
"Carlos"	"Pace"	"1944-10-06"	7.675675675675675
"François"	"Migault"	"1944-12-04"	15.25
"Jacky"	"Ickx"	"1945-01-01"	5.893939393939393
"David"	"Purley"	"1945-01-26"	12.75
"Thomas"	"Monarch"	"1945-09-03"	NULL
"George"	"Eaton"	"1945-11-12"	12.0
"Masahiro"	"Hasemi"	"1945-11-13"	11.0
"Brett"	"Lunger"	"1945-11-14"	11.434782608695652
"Damien"	"Magee"	"1945-11-17"	14.0
"Brian"	"McGuire"	"1945-12-13"	NULL
"Mike"	"Wilds"	"1946-01-07"	NULL
"Alberto"	"Colombo"	"1946-02-23"	NULL
"Hiroshi"	"Fushida"	"1946-03-10"	NULL
"Michel"	"Leclère"	"1946-03-18"	11.6
"John"	"Watson"	"1946-05-04"	6.6
"Jean-Pierre"	"Jarier"	"1946-07-10"	9.126984126984125
"Emilio"	"de Villota"	"1946-07-26"	15.0
"Brian"	"Henton"	"1946-09-19"	9.7
"Jochen"	"Mass"	"1946-09-30"	7.718749999999997
"Alan"	"Jones"	"1946-11-02"	6.235294117647061
"Slim"	"Borgudd"	"1946-11-25"	10.333333333333334
"Emerson"	"Fittipaldi"	"1946-12-12"	6.0212765957446805
"Renzo"	"Zorzi"	"1946-12-12"	9.666666666666666

"Otto"	"Stuppacher"	"1947-03-03"	NULL
"Andy"	"Sutcliffe"	"1947-05-09"	NULL
"Bob"	"Evans"	"1947-06-11"	12.8
"Kazuyoshi"	"Hoshino"	"1947-07-01"	11.0
"Torsten"	"Palm"	"1947-07-23"	10.0
"Ian"	"Scheckter"	"1947-08-22"	11.5
"James"	"Hunt"	"1947-08-29"	4.195652173913045
"Rikky"	"von Opel"	"1947-10-14"	11.5
"Ian"	"Ashley"	"1947-10-26"	15.5
"Giorgio"	"Francia"	"1947-11-08"	NULL
"Roger"	"Williamson"	"1948-02-02"	NULL
"Jim"	"Crawford"	"1948-02-13"	13.0
"Mikko"	"Kozarowitzky"	"1948-05-17"	NULL
"Hans"	"Binder"	"1948-06-12"	10.333333333333332
"René"	"Arnoux"	"1948-07-04"	6.902439024390242
"Harald"	"Ertl"	"1948-08-31"	10.909090909090908
"Jean-Louis"	"Schlesser"	"1948-09-12"	11.0
"Helmuth"	"Koinigg"	"1948-11-03"	10.0
"Alex"	"Ribeiro"	"1948-11-07"	11.0
"Gunnar"	"Nilsson"	"1948-11-20"	7.071428571428572
"Guy"	"Tunmer"	"1948-12-01"	11.0
"Keke"	"Rosberg"	"1948-12-06"	5.847457627118643
"Roelof"	"Wunderink"	"1948-12-12"	NULL
"Niki"	"Lauda"	"1949-02-22"	4.263736263736265
"Ricardo"	"Zunino"	"1949-04-13"	9.666666666666666
"Boy"	"Lunger"	"1949-05-03"	NULL
"Tom"	"Pryce"	"1949-06-11"	7.136363636363637
"Patrick"	"Tambay"	"1949-06-25"	6.24561403508772
"Ricardo"	"Londoño"	"1949-08-08"	NULL
"Michael"	"Bleekemolen"	"1949-10-02"	NULL
"Patrick"	"Nève"	"1949-10-13"	12.375
"Warwick"	"Brown"	"1949-12-24"	14.0
"Gilles"	"Villeneuve"	"1950-01-18"	5.948717948717946
"Gianfranco"	"Brancatelli"	"1950-01-18"	NULL
"Jody"	"Scheckter"	"1950-01-29"	5.194805194805193
"Danny"	"Sullivan"	"1950-03-09"	9.857142857142858
"Larry"	"Perkins"	"1950-03-18"	13.0
"Loris"	"Kessel"	"1950-04-01"	12.0
"Hans-Joachim"	"Stuck"	"1951-01-01"	7.538461538461539
"Geoff"	"Lees"	"1951-05-01"	10.666666666666666
"Noritake"	"Takahara"	"1951-06-06"	9.0
"Marc"	"Surer"	"1951-09-18"	8.980392156862742
"Manfred"	"Winkelhock"	"1951-10-06"	10.153846153846155
"Tiff"	"Needell"	"1951-10-29"	NULL
"Piercarlo"	"Ghinzani"	"1952-01-16"	10.700000000000003
"Patrick"	"Gaillard"	"1952-02-12"	13.0
"Stephen"	"South"	"1952-02-19"	NULL

"Didier"	"Pironi"	"1952-03-26"	6.159090909090908
"Tony"	"Brise"	"1952-03-28"	9.5
"Nelson"	"Piquet"	"1952-08-17"	4.139344262295081
"Bruno"	"Giacomelli"	"1952-09-10"	9.428571428571429
"Bernard"	"de Dryver"	"1952-09-19"	NULL
"Siegfried"	"Stohr"	"1952-10-10"	9.333333333333334
"Bobby"	"Rahal"	"1953-01-10"	12.0
"Dave"	"Kennedy"	"1953-01-15"	NULL
"Satoru"	"Nakajima"	"1953-02-23"	8.514285714285714
"Ingo"	"Hoffmann"	"1953-02-28"	9.0
"Derek"	"Daly"	"1953-03-11"	8.538461538461538
"Lamberto"	"Leoni"	"1953-05-24"	NULL
"Nigel"	"Mansell"	"1953-08-08"	3.9183673469387763
"Miguel Ángel"	"Guerra"	"1953-08-31"	NULL
"Jacques"	"Villeneuve Sr."	"1953-11-04"	NULL
"Desiré"	"Wilson"	"1953-11-26"	NULL
"Jo"	"Gartner"	"1954-01-24"	11.0
"Mauro"	"Baldi"	"1954-01-31"	8.882352941176473
"Riccardo"	"Patrese"	"1954-04-17"	6.43650793650794
"Philippe"	"Alliot"	"1954-07-27"	10.295454545454547
"Oscar"	"Larrauri"	"1954-08-19"	14.5
"Derek"	"Warwick"	"1954-08-27"	7.70149253731343
"Huub"	"Rothengatter"	"1954-10-08"	8.833333333333334
"Eliseo"	"Salazar"	"1954-11-14"	10.5
"Alain"	"Prost"	"1955-02-24"	2.958041958041959
"Rupert"	"Keegan"	"1955-02-26"	10.666666666666666
"Teo"	"Fabi"	"1955-03-09"	6.944444444444444
"Toshio"	"Suzuki"	"1955-03-10"	13.0
"Philippe"	"Streiff"	"1955-06-26"	8.866666666666667
"Johnny"	"Cecotto"	"1956-01-25"	10.285714285714286
"Hector"	"Rebaque"	"1956-02-05"	7.733333333333333
"Kevin"	"Cogan"	"1956-03-31"	NULL
"Jan"	"Lammers"	"1956-06-02"	12.166666666666668
"Stefan"	"Johansson"	"1956-09-08"	6.090909090909091
"Jonathan"	"Palmer"	"1956-11-07"	9.42222222222222
"Michele"	"Alboreto"	"1956-12-23"	7.509803921568627
"François"	"Hesnault"	"1956-12-30"	8.6
"Beppe"	"Gabbiani"	"1957-01-02"	NULL
"Chico"	"Serra"	"1957-02-03"	9.545454545454545
"Thierry"	"Boutsen"	"1957-07-13"	7.412371134020618
"Stefan"	"Bellof"	"1957-11-20"	8.571428571428571
"Kenny"	"Acheson"	"1957-11-27"	12.0
"Raul"	"Boesel"	"1957-12-04"	10.899999999999999
"Eddie"	"Cheever"	"1958-01-10"	7.545454545454545
"Giovanni"	"Lavaggi"	"1958-02-18"	12.5
"Elio"	"de Angelis"	"1958-03-26"	5.796610169491524
"Christian"	"Danner"	"1958-04-04"	8.8

"Johnny"	"Dumfries"	"1958-04-26"	7.166666666666667
"Tommy"	"Byrne"	"1958-05-06"	NULL
"Riccardo"	"Paletti"	"1958-06-15"	NULL
"Olivier"	"Grouillard"	"1958-09-02"	11.142857142857142
"Franco"	"Forini"	"1958-09-22"	NULL
"Roberto"	"Guerrero"	"1958-11-16"	12.2
"Roberto"	"Moreno"	"1959-02-11"	9.764705882352942
"Luis"	"Pérez-Sala"	"1959-05-15"	11.0
"Andrea"	"de Cesaris"	"1959-05-31"	8.338028169014086
"Martin"	"Brundle"	"1959-06-01"	7.192771084337348
"Alessandro"	"Nannini"	"1959-07-07"	6.838709677419354
"Giovanna"	"Amati"	"1959-07-20"	NULL
"Gerhard"	"Berger"	"1959-08-27"	5.096774193548385
"Pascal"	"Fabre"	"1960-01-09"	11.375
"Ayrton"	"Senna"	"1960-03-21"	3.1481481481481484
"Adrián"	"Campos"	"1960-06-17"	15.0
"Roland"	"Ratzenberger"	"1960-07-04"	11.0
"Claudio"	"Langes"	"1960-07-20"	NULL
"Aguri"	"Suzuki"	"1960-09-08"	9.160000000000002
"Damon"	"Hill"	"1960-09-17"	4.480519480519479
"Joachim"	"Winkelhock"	"1960-10-24"	NULL
"Perry"	"McCarthy"	"1961-03-03"	NULL
"Gary"	"Brabham"	"1961-03-29"	NULL
"Mike"	"Thackwell"	"1961-03-30"	NULL
"Corrado"	"Fabi"	"1961-04-12"	9.333333333333334
"Paolo"	"Barilla"	"1961-04-20"	13.0
"Pierluigi"	"Martini"	"1961-04-23"	9.309090909090909
"Pierre-Henri"	"Raphanel"	"1961-05-27"	NULL
"Yannick"	"Dalmas"	"1961-07-28"	11.2
"Allen"	"Berg"	"1961-08-01"	13.666666666666666
"Eric"	"van de Poele"	"1961-09-30"	10.666666666666666
"Julian"	"Bailey"	"1961-10-09"	11.4
"Emanuele"	"Pirro"	"1962-01-12"	9.705882352941176
"Gabriele"	"Tarquini"	"1962-03-02"	10.692307692307693
"Volker"	"Weidler"	"1962-03-18"	NULL
"Michael"	"Andretti"	"1962-10-05"	7.333333333333333
"Bertrand"	"Gachot"	"1962-12-23"	10.066666666666665
"Jean-Marc"	"Gounon"	"1963-01-01"	12.75
"Fabrizio"	"Barbazza"	"1963-04-02"	7.666666666666667
"Maurício"	"Gugelmin"	"1963-04-20"	9.193548387096776
"Paul"	"Belmondo"	"1963-04-23"	12.2
"Stefano"	"Modena"	"1963-05-12"	10.133333333333335
"Ivan"	"Capelli"	"1963-05-24"	8.13793103448276
"Ukyo"	"Katayama"	"1963-05-29"	10.909090909090908
"Taki"	"Inoue"	"1963-09-05"	11.2
"Érik"	"Comas"	"1963-09-28"	9.28125
"Jean-Denis"	"Délétraz"	"1963-10-01"	15.0

"Alex"	"Caffi"	"1964-03-18"	9.384615384615387
"Nicola"	"Larini"	"1964-03-19"	10.807692307692312
"Martin"	"Donnelly"	"1964-03-26"	9.833333333333334
"Andrea"	"Chiesa"	"1964-05-06"	NULL
"Andrea"	"Montermini"	"1964-05-30"	9.75
"Jean"	"Alesi"	"1964-06-11"	6.210084033613445
"Johnny"	"Herbert"	"1964-06-25"	7.722222222222221
"Bernd"	"Schneider"	"1964-07-20"	12.333333333333334
"Éric"	"Bernard"	"1964-08-24"	9.227272727272727
"Enrico"	"Bertaggia"	"1964-09-19"	NULL
"Pedro"	"Chaves"	"1965-02-27"	NULL
"Gregor"	"Foitek"	"1965-03-27"	11.0
"David"	"Brabham"	"1965-09-05"	12.714285714285715
"Marco"	"Apicella"	"1965-10-07"	NULL
"Eddie"	"Irvine"	"1965-11-10"	5.98823529411765
"Jyrki"	"Järvilehto"	"1966-01-31"	8.678571428571429
"Mark"	"Blundell"	"1966-04-08"	7.548387096774195
"Naoki"	"Hattori"	"1966-06-13"	NULL
"Olivier"	"Panis"	"1966-09-02"	8.727272727272723
"Vincenzo"	"Sospiri"	"1966-10-07"	NULL
"Alessandro"	"Zanardi"	"1966-10-23"	10.277777777777779
"Mika"	"Salo"	"1966-11-30"	8.671874999999998
"Heinz-Harald"	"Frentzen"	"1967-05-18"	6.520833333333336
"Domenico"	"Schiattarella"	"1967-11-17"	14.333333333333334
"Gianni"	"Morbidelli"	"1968-01-13"	10.294117647058822
"Emanuele"	"Naspetti"	"1968-02-24"	12.0
"Michael"	"Bartels"	"1968-03-08"	NULL
"Ricardo"	"Rosset"	"1968-07-27"	11.0
"Franck"	"Lagorce"	"1968-09-01"	11.0
"Mika"	"Häkkinen"	"1968-09-28"	4.466019417475731
"Karl"	"Wendlinger"	"1968-12-20"	9.555555555555555
"Michael"	"Schumacher"	"1969-01-03"	3.7012448132780076
"Hideki"	"Noda"	"1969-03-07"	NULL
"Massimiliano"	"Papis"	"1969-10-03"	9.5
"Philippe"	"Adams"	"1969-11-19"	16.0
"Olivier"	"Beretta"	"1969-11-23"	9.5
"Jean-Christophe"	"Boullion"	"1969-12-27"	8.714285714285714
"Allan"	"McNish"	"1969-12-29"	10.875
"Pedro"	"Diniz"	"1970-05-22"	9.358974358974356
"Christian"	"Fittipaldi"	"1971-01-18"	8.96153846153846
"Luca"	"Badoer"	"1971-01-25"	11.6
"Pedro"	"de la Rosa"	"1971-02-24"	12.339285714285715
"David"	"Coulthard"	"1971-03-27"	5.8284023668639
"Shinji"	"Nakano"	"1971-04-01"	10.499999999999998
"Jacques"	"Villeneuve"	"1971-04-09"	7.046296296296295
"Jos"	"Verstappen"	"1972-03-04"	10.53061224489796
"Pedro"	"Lamy"	"1972-03-20"	10.736842105263158

"Rubens"	"Barrichello"	"1972-05-23"	7.056768558951965
"Giancarlo"	"Fisichella"	"1973-01-14"	8.228395061728394
"Jan"	"Magnussen"	"1973-07-04"	9.88888888888889
"Cristiano"	"da Matta"	"1973-09-19"	9.85
"Toranosuke"	"Takagi"	"1974-02-12"	11.666666666666668
"Alexander"	"Wurz"	"1974-02-15"	8.319148936170214
"Marc"	"Gené"	"1974-03-29"	11.347826086956522
"Jarno"	"Trulli"	"1974-07-13"	9.470588235294112
"Norberto"	"Fontana"	"1975-01-20"	10.666666666666666
"Yuji"	"Ide"	"1975-01-21"	13.0
"Luciano"	"Burti"	"1975-03-05"	10.222222222222221
"Gastón"	"Mazzacane"	"1975-05-08"	12.538461538461538
"Ralph"	"Firman"	"1975-05-20"	11.75
"Ralf"	"Schumacher"	"1975-06-30"	6.254098360655738
"Juan"	"Pablo Montoya"	"1975-09-20"	4.209677419354837
"Stéphane"	"Sarrazin"	"1975-11-02"	NULL
"Tarso"	"Marques"	"1976-01-19"	12.7
"Ricardo"	"Zonta"	"1976-03-23"	9.894736842105262
"Alex"	"Yoong"	"1976-07-20"	12.166666666666668
"Tiago"	"Monteiro"	"1976-07-24"	13.344827586206897
"Mark"	"Webber"	"1976-08-27"	6.600000000000001
"Tomáš"	"Enge"	"1976-09-11"	13.0
"Narain"	"Karthikeyan"	"1977-01-14"	17.18181818181818
"Takuma"	"Sato"	"1977-01-28"	11.57627118644068
"Nick"	"Heidfeld"	"1977-05-10"	8.355072463768115
"Franck"	"Montagny"	"1978-01-05"	16.666666666666668
"Nicolas"	"Kiesa"	"1978-03-03"	12.8
"Esteban"	"Tuero"	"1978-04-22"	12.5
"Justin"	"Wilson"	"1978-07-31"	12.571428571428571
"Enrique"	"Bernoldi"	"1978-10-19"	11.444444444444443
"Giorgio"	"Pantano"	"1979-02-04"	14.666666666666666
"Sébastien"	"Bourdais"	"1979-02-28"	12.249999999999998
"Christijan"	"Albers"	"1979-04-16"	14.0
"Anthony"	"Davidson"	"1979-04-18"	14.285714285714285
"Kimi"	"Räikkönen"	"1979-10-17"	5.358565737051793
"Jenson"	"Button"	"1980-01-19"	7.409090909090909
"Markus"	"Winkelhock"	"1980-06-13"	NULL
"Vitantonio"	"Liuzzi"	"1980-08-06"	13.818181818181811
"Antônio"	"Pizzonia"	"1980-09-11"	9.90909090909091
"Patrick"	"Friesacher"	"1980-09-26"	14.4
"Zsolt"	"Baumgartner"	"1981-01-01"	13.5
"Felipe"	"Massa"	"1981-04-25"	7.266375545851529
"Gianmaria"	"Bruni"	"1981-05-30"	15.88888888888889
"Fernando"	"Alonso"	"1981-07-29"	5.87795275590551
"Robert"	"Doornbos"	"1981-09-23"	14.11111111111111
"Heikki"	"Kovalainen"	"1981-10-19"	11.83516483516484
"André"	"Lotterer"	"1981-11-19"	NULL

"Timo"	"Glock"	"1982-03-18"	13.770270270270272
"Sakon"	"Yamamoto"	"1982-07-09"	17.42857142857143
"Adrian"	"Sutil"	"1983-01-11"	12.471910112359552
"Scott"	"Speed"	"1983-01-24"	12.294117647058824
"Christian"	"Klien"	"1983-02-07"	11.228571428571428
"Bruno"	"Senna"	"1983-10-15"	13.885714285714286
"Karun"	"Chandhok"	"1984-01-19"	17.22222222222222
"Lucas"	"di Grassi"	"1984-08-11"	17.6
"Vitaly"	"Petrov"	"1984-09-08"	12.914893617021276
"Robert"	"Kubica"	"1984-12-07"	9.289156626506026
"Lewis"	"Hamilton"	"1985-01-07"	3.36283185840708
"Kazuki"	"Nakajima"	"1985-01-11"	12.2
"Pastor"	"Maldonado"	"1985-03-09"	12.805970149253733
"Giedo"	"van der Garde"	"1985-04-25"	17.375000000000004
"Nico"	"Rosberg"	"1985-06-27"	6.355932203389829
"Nelson"	"Piquet Jr."	"1985-07-25"	10.470588235294118
"Jérôme"	"d'Ambrosio"	"1985-12-27"	17.882352941176475
"Paul"	"di Resta"	"1986-04-16"	10.576923076923078
"Romain"	"Grosjean"	"1986-04-17"	10.713114754098362
"Kamui"	"Kobayashi"	"1986-09-13"	10.849056603773585
"Sebastian"	"Vettel"	"1987-07-03"	3.942857142857142
"Nico"	"Hülkenberg"	"1987-08-19"	9.549295774647888
"Sébastien"	"Buemi"	"1988-10-31"	11.975
"Daniel"	"Ricciardo"	"1989-07-01"	7.9999999999999964
"Jules"	"Bianchi"	"1989-08-03"	16.57142857142857
"Valtteri"	"Bottas"	"1989-08-28"	6.179687500000001
"Brendon"	"Hartley"	"1989-11-10"	13.352941176470589
"Sergio"	"Pérez"	"1990-01-26"	9.312101910828027
"Charles"	"Pic"	"1990-02-15"	17.299999999999997
"Jaime"	"Alguersuari"	"1990-03-23"	12.08333333333333
"Jean-Éric"	"Vergne"	"1990-04-25"	11.911111111111115
"Marcus"	"Ericsson"	"1990-09-02"	13.932432432432432
"Jolyon"	"Palmer"	"1991-01-20"	13.26923076923077
"Roberto"	"Merhi"	"1991-03-22"	15.499999999999998
"Max"	"Chilton"	"1991-04-21"	17.343749999999993
"Will"	"Stevens"	"1991-06-28"	16.125
"Esteban"	"Gutiérrez"	"1991-08-05"	13.956521739130439
"Alexander"	"Rossi"	"1991-09-25"	15.4
"Stoffel"	"Vandoorne"	"1992-03-26"	12.117647058823527
"Felipe"	"Nasr"	"1992-08-21"	13.090909090909092
"Kevin"	"Magnussen"	"1992-10-05"	11.609195402298855
"Rio"	"Haryanto"	"1993-01-22"	18.22222222222222
"Antonio"	"Giovinazzi"	"1993-12-14"	13.571428571428571
"Daniil"	"Kvyat"	"1994-04-26"	10.621621621621628
"Carlos"	"Sainz"	"1994-09-01"	9.233766233766232
"Pascal"	"Wehrlein"	"1994-10-18"	14.999999999999998
"Sergey"	"Sirotkin"	"1995-08-27"	14.88888888888889

```
| "Pierre"          | "Gasly"               | "1996-02-07" | 10.390243902439027 |
| "Alexander"       | "Albon"               | "1996-03-23" | 8.85               |
| "Esteban"         | "Ocon"                | "1996-09-17" | 10.302325581395348 |
| "Max"             | "Verstappen"          | "1997-09-30" | 5.07317073170732   |
| "Charles"         | "Leclerc"             | "1997-10-16" | 6.852941176470588  |
| "George"          | "Russell"             | "1998-02-15" | 15.526315789473685 |
| "Lance"           | "Stroll"              | "1998-10-29" | 12.444444444444443 |
| "Lando"           | "Norris"              | "1999-11-13" | 9.411764705882353  |
+-------------------------------------------------------------------------------------+
```