

# CS 451 Final Project: Formula One Database

Zane Globus-O'Harra

*23 March 2023*

## Contents

<b>1</b>	<b>Cover Page</b>	<b>1</b>
<b>2</b>	<b>Summary</b>	<b>1</b>
2.1	High Level Overview . . . . .	1
2.2	Kinds of Data . . . . .	1
2.3	Application Programs . . . . .	2
<b>3</b>	<b>Logical Design</b>	<b>2</b>
<b>4</b>	<b>Physical Design</b>	<b>2</b>
<b>5</b>	<b>List of Applications</b>	<b>3</b>
<b>6</b>	<b>User's Guide</b>	<b>4</b>
6.1	Basic Overview . . . . .	4
6.2	Examples of Each Application . . . . .	5
<b>7</b>	<b>Contents of the Tables</b>	<b>7</b>
<b>8</b>	<b>Implementation Code</b>	<b>7</b>
<b>9</b>	<b>Conclusion</b>	<b>7</b>
9.1	Reflection About the Project . . . . .	7
9.2	Simplifications . . . . .	8
9.3	Things I Would Have Done Differently . . . . .	9

NOTE: All links in this PDF should be clickable.

# 1 Cover Page

- **Author:** Zane Globus-O’Harra
- **Project Title:** Formula One Database
- **Connection Information:**
  - **Port Number:** 3372
  - **Host Name:** ix.cs.uoregon.edu
  - **Guest Account Login & Password:**  
login: guest  
password: guest
  - **Database Name:** f1db
- **Project URL:** <https://ix.cs.uoregon.edu/~zfg/f1db/main.html>
- **Highlights:**
  - Population of the result table (440 rows).
    - \* The population of all tables except for the season table (1 row) and team table (10 rows) have at least 20 rows.
  - 8 applications:
    - \* 6 applications to display driver and team results in different and meaningful ways.
    - \* 1 application to display race information *without* driver or team results (i.e., race name, circuit name, race length, number of laps, etc.).
    - \* 1 application to compare the drivers of a team.

## 2 Summary

### 2.1 High Level Overview

The world that will be modeled will be a simplified version of a Formula One (Formula 1, or F1) car-racing season. This will include the drivers, the teams the drivers are in, the races, and the results of those races.

### 2.2 Kinds of Data

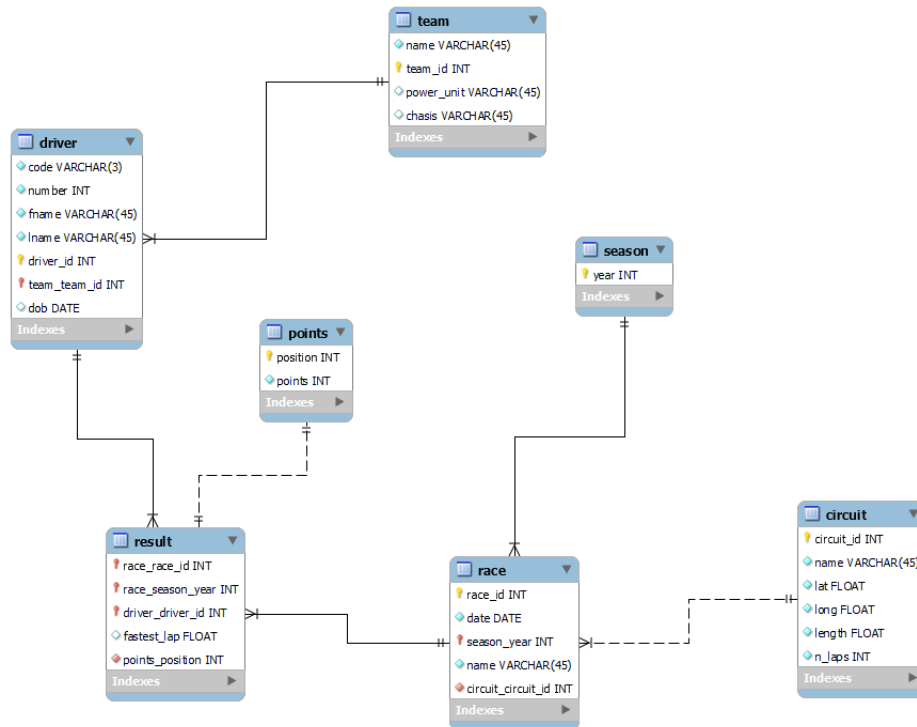
The kinds of data that will be stored will be about the drivers, the teams, and the results that those drivers receive in the races that they participate in. I’m not sure how in-depth you want me to go when discussing kinds of data, but I think that the high level overview and the ER diagram in the Logical Design section make it fairly clear the kinds of data that I will be keeping track of.

## 2.3 Application Programs

The application programs that are desired are a way to summarize the results in a season, and determine who was the champion for that season. I also want to look at the average results of each driver, and give a summary of their best and worst races in a season (if I add data for multiple seasons, then I will also be able to look at their results across their career).

Each team will have two drivers, and I will have a way to compare the drivers of one team. I will also have a way to look at the results of a team over a season (the team's results is the sum of the results of its drivers), and the results of that team over its tenure in F1.

## 3 Logical Design



## 4 Physical Design

Table names are in **bold and CAPITALIZED**, primary keys are underlined, and foreign keys are *italicized*. If an attribute is a primary key and a foreign key, then

it is *underlined and italicized*. Not null requirements are indicated. Primary keys and foreign keys are always not null.

- **TEAM:** team\_id, name, power\_unit, chassis
  - not null: power\_unit, chassis
- **DRIVER:** driver\_id, *team\_team\_id*, code, number, fname, lname, dob
  - not null: code, number, fname, lname
  - *team\_team\_id* references **TEAM**
- **RESULT:** race\_race\_id, race\_season\_year, driver\_driver\_id, *points\_position*, fastest\_lap
  - race\_race\_id references **RACE**
  - race\_season\_year references **SEASON**
  - driver\_driver\_id references **DRIVER**
  - *points\_position* references **POINTS**
- **POINTS:** position, points
  - not null: points
- **RACE:** race\_id, season\_year, circuit\_circuit\_id, date, name
  - not null: date, name
  - season\_year references **SEASON**
  - circuit\_circuit\_id references **CIRCUIT**
- **SEASON:** year
- **CIRCUIT:** circuit\_id, name, lat, long, length, n\_laps
  - not null: name, lat, long, length, n\_laps

## 5 List of Applications

- (1) input: season  
output: summary of the all driver's results across that season
- (2) input: season  
output: summary of the team (constructor) results of that season
- (3) input: race, season  
output: summary of the driver's results from that one race in that season

- (4) input: race, season  
output: summary of the team's results from that race in that season
- (5) input: driver, season  
output: A single driver's race results across that season
- (6) input: driver, season  
output: average results of that driver across that season
- (7) input: season  
output: summary of the races in that season (race name, circuit name, circuit length, number of laps, etc.)
- (8) input: team, season  
output: for each race in that season, compare the results of the drivers in that team

## 6 User's Guide

### 6.1 Basic Overview

Each “application” is fairly straightforward to use. Most of the applications display some variation of the results over a season. Seasons are determined by the year, and the only season that has data is 2021. However, the complete data for every race and every result has been entered for the 2021 F1 season, so every query for that season should return a fairly “well-stocked” set of results.

The other fields are fairly picky about what is entered. If a field asks for a “Race,” it wants that race's full name. The available races are:

- |                             |                            |
|-----------------------------|----------------------------|
| • Bahrain Grand Prix        | • Belgian Grand Prix       |
| • Emilia Romagna Grand Prix | • Dutch Grand Prix         |
| • Portugese Grand Prix      | • Italian Grand Prix       |
| • Spanish Grand Prix        | • Russian Grand Prix       |
| • Monaco Grand Prix         | • Turkish Grand Prix       |
| • Azerbaijan Grand Prix     | • United States Grand Prix |
| • French Grand Prix         | • Mexico City Grand Prix   |
| • Styrian Grand Prix        | • Sao Paulo Grand Prix     |
| • Austrian Grand Prix       | • Qatar Grand Prix         |
| • British Grand Prix        | • Saudi Arabian Grand Prix |
| • Hungarian Grand Prix      | • Abu Dhabi Grand Prix     |

The same goes for drivers, whose last names are:

- Raikkonen
- Giovinazzi
- Gasly
- Tsunoda
- Alonso
- Ocon
- Vettel
- Stroll
- Leclerc
- Sainz
- Mazepin
- Schumacher
- Ricciardo
- Norris
- Hamilton
- Bottas
- Perez
- Verstappen
- Latifi
- Russell
- Kubica

And lastly, the same goes for the teams:

- Mercedes-AMG Petronas F1 Team
- Red Bull Racing Honda
- Scuderia Ferrari
- McLaren F1 Team
- Aston Martin Cognizant F1 Team
- Alpine F1 Team
- Alfa Romeo Racing Orlen
- Scuderia AlphaTauri Honda
- Haas F1 Team
- Williams Racing

## 6.2 Examples of Each Application

Examples for each app are as follows:

Enter a season to display the final driver standings of that season.

Season:

Figure 1: This app displays the final driver standings of a season, by summing up the points that each driver has earned over the season and ordering the drivers based on total points earned.

Enter a season to display the final team standings of that season.

Season:

Figure 2: This app displays the final team standings of a season, by summing up the points that each team has earned over the season and ordering the team based on total points earned. Because each team has two drivers, the points that a team earns in a race is the sum of the points of the drivers of that team.

Enter a race and a season to display the driver standings for that race in that season.

Race:   
Season:

Figure 3: This app displays the driver standings after a race, showing their name, the position they finished in, and the points that they earned.

Enter a race and a season to display the team standings for that race in that season.

Race:   
Season:

Figure 4: This app displays the team standings after a race, showing their name, the position they finished in, and the points that they earned. The teams finishing position is determined by the points that the drivers of that team earned.

Enter a driver and a season to display that driver's race results across the season.

Driver Last Name:   
Season:

Figure 5: This app displays, in chronological order across a season, the race results of one driver as well as the points that that driver earned.

Enter a season to display the average results of every driver over that season.

Season:

Figure 6: This app displays the average position per race and average points per race for every driver in a season, ordered from the highest average points to the lowest.

Enter a season to display information about each race of the season.

Season:

Figure 7: This app displays information about every race that occurs in a season, including the race name, the circuit at which the race takes place, the date of the race, the length of the circuit, the number of laps, and the total length of the race. The results are ordered chronologically by race date.

Enter a team name and a season to display information about that team's drivers during that season.

Team Name:   
Season:

Figure 8: This app displays the two drivers in a team, mainly to compare the number of points that each driver has earned, but also showing their driver codes, their number, and the number of races they have completed. The drivers' ages are also displayed, and it should be noted that these are *not* the drivers' current ages, rather they are the drivers' ages during the specified season. If a needed a replacement driver for some races during the season, that is displayed here (as is the case with Alfa Romeo).

## 7 Contents of the Tables

The contents of the tables can be found by clicking on the link to the table contents from the main project URL, or by going directly to:  
<https://ix.cs.uoregon.edu/~zfg/f1db/tables.html>

## 8 Implementation Code

The implementation code for the main website can be found via a link at the bottom of the main project webpage. The code for each PHP app can be found via a link below each submission form. The link to the main website is repeated here:  
<https://ix.cs.uoregon.edu/~zfg/f1db/main.html>

## 9 Conclusion

### 9.1 Reflection About the Project

Looking back on what I have done in this project, I would say that the bulk of the work was split into three parts:



- (1) Designing the database.
- (2) Entering data.
- (3) Creating the applications.

After creating my initial Chen diagram for part 1 of this project, I revised it and updated it for part 2 into a Crowsfoot diagram made in MySQL workbench so that I could easily turn the ER diagram into an actual database with tables. This part took a while because of the planning and thinking that was involved around how to organize the data, as well as what data I needed for the applications that I wanted to make.

Once I had the design that I wanted, I then needed to enter the data. I decided to enter the data for one entire Formula 1 season. While I could have easily used the more recent 2022 season data, I decided to instead use the 2021 season data because that was a very exciting season with many thrilling races. The data that I used was from Wikipedia,<sup>1</sup> and I could find no easy way to enter it in bulk (I'm sure if I searched harder I could have found a way, but oh well), so I entered it all manually. For the most part this wasn't too tedious, except for the results table, which was 440 rows.

Once I had the data entered, I set about creating the applications. I already had an idea about what applications I wanted to make (my original application ideas are still listed in section 2.3), so I started with those ideas. Because I only had the data for one season, this meant that I couldn't compare a driver's results across seasons, so I had to change some of my applications to account for this. Additionally, I found it exceptionally difficult to create a query that would tell me the best and worst results of a driver, so I ended up abandoning that.

Instead, I added applications to display information about each race of a season, as well as some additional ways to display the results of each driver or team on a per race or per season basis.

I also want to add some of my reasoning behind adding an ID field to many of the tables. While drivers could have been ID'd by their code, if I had added previous seasons, then there would be an overlap between the driver code of "MSC" for Mick Schumacher and his father, Michael Schumacher, who also raced in F1. There is similar reasoning for the other tables (e.g., races change names depending on the season, and so do the teams' names, and so on).

## 9.2 Simplifications

While my database captures a great deal of the scoring system and results system of Formula 1, it has nevertheless been simplified. Some things that my database simplifies are:

---

<sup>1</sup>The data was mainly from [wikipedia.org/wiki/2021\\_Formula\\_One\\_World\\_Championship](https://wikipedia.org/wiki/2021_Formula_One_World_Championship), as well as the Wikipedia pages associated with each driver, race, and circuit.

- Tiebreakers: When two drivers have the same score at the end of a season, their race results are “counted back,” and the driver with higher ranking finishes wins the tie. There is no such thing in my database, and ties are broken by the alphabetical order of drivers’ names.
- Fastest laps: When a driver finishes a race in the top 10 and has completed the fastest lap in that race, they earn an additional point. My database includes no such rule.
- Partially awarded points: When a race does not complete the full race distance, half points are awarded. This occurred in the 2021 Belgian Grand Prix due to poor weather conditions. My database does not have a rule to allow this.
- Sprint races: I won’t go into the details of what a sprint race entails, but it adds another layer of complexity to the point system.
- Qualifying: A qualifying session precedes every race to determine the order of the starting grid. Ideally my database would include the results of this qualifying session as well, because then I could create a query that could examine the positions gained or lost in the race relative to a driver’s starting position.

The list goes on, but these are the main areas that my database has simplified.

### 9.3 Things I Would Have Done Differently

Aside from the simplifications mentioned in section 9.2, I would have done two main things differently:

- Change fastest laps: As they currently are, fastest laps have only one column and they are in units of minutes. I think that it would have been better to split fastest lap into two columns, one for the minutes and one for the seconds. While none of my applications involved fastest laps, I think that this would have made queries involving fastest laps easier to write (i.e., I wouldn’t have to convert 1.5 minutes to 1 minute and 30 seconds because it would already be in the minute-second format).
- Have an additional table for engines: In the team table, I have the engine that each team uses as a column. However, some teams buy engines from other teams, and thus use the same engine (e.g., Mercedes-AMG Petronas F1 Team produces the Mercedes-AMG F1 M12 engine, which is also used by McLaren F1 Team, Aston Martin Cognizant F1 Team, and Williams Racing). This violates the “one fact in one place” motto.
- Have a better system for calculating a team’s position results based on one race. As it is, I use the `RANK() OVER` function in MySQL, which ranks each team based on the sum of the points that the drivers earned in a race. However, if two teams earned the same number of points, then their position is the same.

This does not adhere to real life, where driver positions come into play when teams have the same number of points. I would have liked to implement this, but there were a great deal of rules that are tricky, and I was struggling to get everything figured out. So instead I used a simplified version.

Other than those two small changes, I am quite happy with how this project turned out, and I think that despite being somewhat simple, my database does a fairly good job at modeling a Formula 1 season.