Project team 14 - <Formula One Database>

Team Members: Alyssa Day, Hiza Mvuendy, Joe Posillico, Fabiola Rojas

## PROJECT PROPOSAL

**Content, Scope and Objectives**

We plan on working on a Formula One racing database. This database will store important information regarding the Formula One circuit such as teams, drivers, race locations, previous race results, and seasonal standings, providing a centralized resource for tracking and analyzing key aspects of the sport. A Formula One racing database will be a great tool for team managers and engineers, helping them keep track of driver performances, and race results. It will also allow teams to analyze key details from previous races, helping them make better decisions to improve their future performance. Additionally, this database can be used to gain insights that could benefit teams throughout the season.

This project aims to create a comprehensive Formula One database that captures the relationships between teams, drivers, races, circuits, and results. It will focus on key aspects of the sport, such as team and driver performance, race history, and circuit details, offering a solid foundation for in-depth analysis.

Our goal is to build a structured database that supports efficient data queries and enables insights into Formula One. By tracking race results, driver stats, team standings, and circuit characteristics, we aim to analyze trends across seasons and gain valuable skills in database design and optimization.

This database will be a valuable tool for analysts, teams, and fans to assess performance, compare circuits, and explore seasonal trends. Ultimately, it will provide a systematic way to explore Formula One data, with potential for future predictive analytics and performance improvements.

## PROJECT ENVIRONMENT

The Formula One Racing Database will be developed using MySQL, one of the most widely used relational database management systems globally. Known for its efficiency and reliability, MySQL utilizes SQL (Structured Query Language) to manage, query, and manipulate data, making it an ideal choice for structuring and maintaining our Formula One racing database. This setup will support essential functions such as data organization, storage, and retrieval, crucial for handling complex relationships between teams, drivers, circuits, race results, and seasonal standings. MySQL’s scalability and robust performance also make it suitable for future expansions, such as incorporating advanced analytics and predictive modeling capabilities. This environment will ensure that our database meets the project’s demands for accuracy, speed, and organization.

**F1 Database**

**Below are the entities involved:**

* **Driver**
  + Contract (Team to Driver) (1-M)
  + Performance Metrics (Race Results to Driver) (M-1)
  + Sponsorships (Driver to Team) (M-1)
  + Ranking (Season Standings to Driver) (M-1)
* **Team Principal**
  + Manages Team (Team Principal to Team) (1-1)
  + Oversees Race Strategy (Team Principal to Race Results) (1-M)
  + Oversees Drivers during race (Team Principal to Driver) (1-M)
* **Team**
  + Drivers (Team to Driver) (1-M)
  + Team Principal (Team to Team Principal) (1-1)
  + Season Ranking (Season Standings to Team) (M-1)
* **Circuit**
  + Races Held (Circuit to Race Results) (1-M)
  + Track Details (Circuit to Team) (1-M)
  + Lap Times (Race Results to Circuit) (M-1)
* **Race Results**
  + Driver Performance (Race Results to Driver) (M-1)
  + Team Performance (Race Results to Team) (M-1)
  + Race Location (Race Results to Circuit) (M-1)
* **Season Standings**
  + Driver Ranking (Season Standings to Driver) (M-1)
  + Team Ranking (Season Standings to Team) (M-1)
* **Sponsors**
  + Sponsorship Deals (Sponsor to Team) (M-M)
  + Money (Sponsor to Team) (M-M)

## HIGH LEVEL REQUIREMENTS

### Initial user roles

| **User Role** | **Description** |
| --- | --- |
| Driver | The one driving the vehicle during the F1 race. |
| Team\_Principle | The manager responsible for overseeing all aspects of an F1 team, including driver performance, race strategy, sponsorships, and team operations. The team principal ensures the team performs competitively. |
| Team | Represents an F1 team consisting of drivers, engineers, and other staff. The team competes in races throughout the season and is ranked in the season standings based on cumulative performance. |
| Circuit | The physical race track where F1 races are held. Each track has its own unique location as well as unique characteristics such as distance. |
| Race\_Results | The results from each F1 race, including driver placements, amount of points awarded to drivers, best lap times. This will track individual driver and team performances |
| Season\_Standings | The overall ranking of drivers and teams based on cumulative performance across all races in a season. This entity tracks both driver and team points earned throughout the season. |

### Initial user story descriptions

| **Story ID** | **Story description** |
| --- | --- |
| US1 | As a Driver, I want to view my race results and lap times so that I can assess my performance in each race. |
| US2 | As a Team Principal, I want to manage driver and team standings so that I can make informed decisions about race strategies and team operations. |
| US3 | As a Team, I want to track overall performance across the season so that I can understand our progress and adjust our strategies accordingly. |
| US4 | As a Circuit Manager, I want to provide detailed race track data to teams so that they can plan race strategies effectively. |
| US5 | As a Race Official, I want to record race results accurately so that we maintain fair competition and transparency. |
| US6 | As a Fan, I want to view race results and season standings so that I can follow my favorite teams and drivers. |
| US7 | As an Engineer, I want to analyze race data from circuits so that I can optimize car performance and make adjustments for future races. |

## HIGH LEVEL CONCEPTUAL DESIGN

1. **Driver**

1.1. Reserve Driver

1.2. Full-time Driver

1. **Team Principal**

2.1. Head of Strategy  
2.2. Operations Manager

1. **Team**

3.1. Racing Team (i.e, Mercedes, Ferrari, Red Bull)

1. **Circuit**

4.1. Permanent Circuit  
4.2. Temporary Street Circuit

1. **Race Results** 5.1. Driver Lap Times  
   5.2. Team Performance Metrics
2. **Season Standings**

6.1. Driver Championship Standings  
6.2. Constructor Championship Standings

1. **Sponsor**

7.1. Title Sponsor  
7.2. Technical Partner

### Relationships:

* Team Principal - manages - Team
* Team Principal - oversees - Driver
* Driver - drives for - Team
* Team - participates in - Race Results
* Race Results - updates - Season Standings
* Driver - competes on - Circuit
* Sponsor - sponsors - Team
* Circuit - hosts - Race Results
* Race Results - influences - Season Standings
* Driver - contributes to - Team Performance
* Team - submits lap times to - Race Results
* Sponsor - provides financial support to - Team

# Sprint 1

## REQUIREMENTS

Refine the user stories that you made in previous sprint. List your updated user stories and any notes you wish to include in decreasing order of priority and highlight the stories chosen for Sprint 1. *There is no need to show your story refinement process - just the list of updated stories suffices.* Use the format shown below.

| **Story ID** | **Story description** |
| --- | --- |
| US1 | As a <role>, I want to <need/feature> so that <reason/benefit> |
| ... | ... |

## CONCEPTUAL DESIGN

Include your detailed conceptual design here. Use the format shown below.

Entity: **Entity1**

Attributes:

attr1\_a

attr1\_b [composite]

part\_1

part\_2

Entity: **Entity2**

Attributes:

attr2\_a

attr2\_b [multi-valued]

attr2\_c [derived]

Relationship: **Entity1** relationship-phrase **Entity2**

Cardinality: <One/Many> to <One/Many>

Participation:

Entity1 has <partial/total> participation

Entity2 has <partial/total> participation

## LOGICAL DESIGN

Include your logical design here. Use the format shown below.

Table: **Table1**

Columns:

pk\_1

column\_1a

column\_1b

*Justification (if needed)*

Table: **Table2**

Columns:

pk\_2

column\_2a

column\_2b [foreign key; references **pk\_1** of **Table1**]

*Justification (if needed)*

## SQL QUERIES

List at least **three** SQL queries that perform data retrievals relevant to the features chosen in the current sprint. For each query, paste a **screenshot** of the output, as shown through database management tool.

Sprint 2

## REQUIREMENTS

Refine the user stories that you made in previous sprint. List your updated user stories in decreasing order of priority. Highlight the stories for which database design was completed in Sprint 1 in one color. Highlight the updated/new stories chosen for Sprint 2 in a different color. *There is no need to explicitly show your story refinement process.* Use the format shown below.

| **Story ID** | **Story description** |
| --- | --- |
| US1 | As a <role>, I want to <need/feature> so that <reason/benefit> |
| ... | ... |

## CONCEPTUAL DESIGN

Include your complete updated conceptual design here. Use the format shown below.

Entity: **Entity1**

Attributes:

attr1\_a

attr1\_b [composite]

part\_1

part\_2

Entity: **Entity2**

Attributes:

attr2\_a

attr2\_b [multi-valued]

attr2\_c [derived]

Relationship: **Entity1** relationship-phrase **Entity2**

Cardinality: <One/Many> to <One/Many>

Participation:

Entity1 has <partial/total> participation

Entity2 has <partial/total> participation

## LOGICAL DESIGN WITH NORMAL FORM IDENTIFICATION

Include your complete updated logical design here. Use the format shown below.

Table: **Table1**

Columns:

pk\_1

column\_1a

column\_1b

*Justification of primary key (if needed)*

Highest normalization level: <1NF/2NF/3NF/BCNF>

Justification (if below BCNF):

Table: **Table2**

Columns:

pk\_2

column\_2a

column\_2b [foreign key; references **pk\_1** of **Table1**]

*Justification of primary key (if needed)*

Highest normalization level: <1NF/2NF/3NF/BCNF>

Justification (if below BCNF):

## SQL QUERIES

Refine your SQL queries that you designed in the previous sprint if in need. List at least **three** SQL queries that perform data retrievals relevant to the features chosen in the current sprint. For each query, paste a **screenshot** of the output, as shown through your user interface.

Sprint 3

## REQUIREMENTS

Refine the user stories that you made in previous sprint. List your updated user stories in decreasing order of priority. Highlight the stories that were completed in Sprint 1 in one color. Highlight the stories that were completed in Sprint 2 in a different color. Highlight the updated/new stories chosen for Sprint 3, if any, in a third color. *There is no need to explicitly show your story refinement process.* Use the format shown below.

| **Story ID** | **Story description** |
| --- | --- |
| US1 | As a <role>, I want to <need/feature> so that <reason/benefit> |
| ... | ... |

## CONCEPTUAL DESIGN

Include your complete updated conceptual design here. Use the format shown below.

Entity: **Entity1**

Attributes:

attr1\_a

attr1\_b [composite]

part\_1

part\_2

Entity: **Entity2**

Attributes:

attr2\_a

attr2\_b [multi-valued]

attr2\_c [derived]

Relationship: **Entity1** relationship-phrase **Entity2**

Cardinality: <One/Many> to <One/Many>

Participation:

Entity1 has <partial/total> participation

Entity2 has <partial/total> participation

## LOGICAL DESIGN WITH HIGHEST NORMAL FORMS AND INDEXES

Include your complete updated logical design here. Use the format shown below.

Table: **Table1**

Columns:

pk\_1

column\_1a

column\_1b

*Justification of primary key (if needed)*

Highest normalization level: <1NF/2NF/3NF/BCNF>

Justification (if below BCNF):

Indexes:

Index #: <type (clustered/non-clustered)>

Columns: <ordered list of columns forming the index>

Justification:

Table: **Table2**

Columns:

pk\_2

column\_2a

column\_2b [foreign key; references **pk\_1** of **Table1**]

*Justification of primary key (if needed)*

Highest normalization level: <1NF/2NF/3NF/BCNF>

Justification (if below BCNF):

Indexes:

Index #: <type (clustered/non-clustered)>

Columns: <ordered list of columns forming the index>

Justification:

## VIEWS AND STORED PROGRAMS

List the views relevant to your application here. Use the format specified below.

**View**: <name of view>

Goal: <1-2 sentence description of what the view contains and what its purpose is (e.g., why and what user(s) would use it, etc.)>

List the stored programs relevant to your application thus far here. Use the format specified below for the different kinds of stored programs. Note: if you do not have a particular type of stored program in your application, just leave that part out.

**Stored procedure**: <name of procedure>

Parameters: <list of parameters, specifying IN/OUT/INOUT for each>

Goal: <1-2 sentence description of what the stored procedure does>

**Stored function**: <name of function>

Parameters: <list of parameters>

Goal: <1-2 sentence description of what the stored function does and what it returns>

**Trigger**: <type of trigger> on <table name>

Goal: <1-2 sentence description of what the trigger does>

**Event**: <type of event>

Goal: <1-2 sentence description of what the event does>