KAGGLE DATASET

**Formula 1 World Championship History (1950-2024)**

## **About Dataset**

The **Formula 1 World Championship History (1950-2024)** dataset offers a comprehensive exploration of Formula 1 racing, spanning over seven decades of competition. This dataset is designed to be a valuable resource for those interested in the evolution of Formula 1, covering every aspect of the sport from its inception in 1950 through the latest 2024 season.

## **Contents of the Dataset**

The dataset is organized into multiple CSV files, each offering detailed data on different aspects of Formula 1:

1) **Track\_Information.csv**: Contains details about the circuits where races have been held, including their location, length, and unique characteristics.

2) **Team\_Details.csv**: Provides information about the constructors, including their history, achievements, and performance across different seasons.

3) **Constructor\_Performance.csv**: Details the performance of constructors in individual races, showing how teams have evolved over the years.

4) **Constructor\_Rankings.csv**: Offers annual standings of constructors, highlighting the competitive dynamics within the sport.

5) **Driver\_Details.csv**: Includes comprehensive information about the drivers, such as their personal details, career statistics, and achievements.

6) **Driver\_Rankings.csv**: Displays the yearly standings of drivers, showcasing who topped the charts and how tight the championship battles were.

7) **Race\_Schedule.csv**: Lists all the races held from 1950 to 2024, along with details such as the date, location, and race name.

8) **Race\_Results.csv**: Provides detailed results of each race, including finishing positions, points earned, and other key metrics.

9) **Lap\_Timings.csv**: Contains data on lap times recorded by drivers during races, giving insights into their performance consistency.

10) **Pit\_Stop\_Records.csv**: Offers information on pit stops made during races, including timing and strategy, which often impact the outcome of the race.

11) **Qualifying\_Results.csv**: Details the results of qualifying sessions, which determine the starting grid for each race.

12) **Sprint\_Race\_Results.csv**: Includes data on sprint races, shorter races introduced to decide starting positions for the main race.

13) **Season\_Summaries.csv**: Summarizes each season, including the number of races, champions, and key moments.

14) **Race\_Status.csv**: Contains codes and descriptions related to the status of cars during a race, such as if a car finished, retired, or was disqualified.

**MySQL**

### 1. **Driver Performance**

**Q1. Career statistics:**

A. Calculate total races –

SELECT COUNT(\*) AS total\_races FROM race\_results;

B. Calculate Wins –

SELECT sum(dr.wins) AS wins, dd.forename, dd.surname

FROM driver\_rankings AS dr

INNER JOIN driver\_details AS dd

ON

dr.driverId = dd.driverId

GROUP BY dr.wins, dd.forename, dd.surname

ORDER BY wins DESC limit 5;

C. Race status analysis –

SELECT rr.driverId, rr.position, rr.points, rs.statusId, rs.status

from race\_status AS rs

INNER JOIN race\_results AS rr

ON

rs.statusId = rr.statusId

GROUP BY rr.driverId, rr.position,rr.points, rs.statusId, rs.status

ORDER BY rr.points AND rr.position DESC LIMIT 10;.

**Q2.Head-to-head comparisons:** Compare the performance of different drivers against each other - SELECT

driver1.forename AS driver1\_name,

driver2.forename AS driver2\_name,

COUNT(CASE WHEN rr1.position < rr2.position THEN 1 END) AS driver1\_wins,

COUNT(CASE WHEN rr2.position < rr1.position THEN 1 END) AS driver2\_wins,

COUNT(CASE WHEN rr1.position <= 3 AND rr2.position <= 3 THEN 1 END) AS total\_podiums,

SUM(CASE WHEN rr1.position = 1 AND rr2.position <= 3 THEN 1 END) AS driver1\_podiums,

SUM(CASE WHEN rr2.position = 1 AND rr1.position <= 3 THEN 1 END) AS driver2\_podiums,

SUM(CASE WHEN rr1.position = 1 AND rr2.position = 2 THEN 1 END) AS driver1\_second\_places,

SUM(CASE WHEN rr2.position = 1 AND rr1.position = 2 THEN 1 END) AS driver2\_second\_places,

SUM(CASE WHEN rr1.position = 1 AND rr2.position = 3 THEN 1 END) AS driver1\_third\_places,

SUM(CASE WHEN rr2.position = 1 AND rr1.position = 3 THEN 1 END) AS driver2\_third\_places

FROM

driver\_details AS driver1

INNER JOIN race\_results AS rr1 ON driver1.driverId = rr1.driverId

INNER JOIN driver\_details AS driver2 ON driver1.driverId <> driver2.driverId

INNER JOIN race\_results AS rr2 ON driver2.driverId = rr2.driverId

WHERE

rr1.raceId = rr2.raceId

GROUP BY

driver1.forename,

driver2.forename

ORDER BY

driver1\_wins DESC,

driver2\_wins DESC

LIMIT 5;

**Q3.Performance trends:** Analyze how drivers' performance has evolved over time. – SELECT

dd.forename,

rs.year,

AVG(rr.position) AS average\_position

FROM

driver\_details AS dd

INNER JOIN race\_results AS rr ON dd.driverId = rr.driverId

INNER JOIN race\_schedule AS rs ON rr.raceId = rs.raceId

GROUP BY

dd.forename,

rs.year

ORDER BY rs.year DESC LIMIT 10;

### 2. **Team Performance**

**Q4.Constructor championships:** Identify the most successful constructors and their dominance. - SELECT td.constructorId, td.name, td.name, td.url,

rr.rank, rr.position, cr.wins

FROM team\_details AS td

INNER JOIN constructor\_rankings AS cr

ON

td.constructorId = cr.constructorId

INNER JOIN race\_results AS rr

ON

td.constructorId = rr.constructorId

GROUP BY

td.constructorId, td.name, td.name, td.url,

rr.rank, rr.position, cr.wins

ORDER BY rr.rank ASC;

**Q5.What is the average number of points scored by drivers per season, and how has this changed over the decades?**

SELECT

CONCAT(FLOOR(race\_schedule.year / 10) \* 10, 's') AS decade,

ROUND(AVG(driver\_season\_points.total\_points), 2) AS avg\_points\_per\_driver

FROM

(SELECT

driver\_rankings.driverId,

race\_schedule.year,

SUM(driver\_rankings.points) AS total\_points

FROM

driver\_rankings

JOIN race\_schedule ON driver\_rankings.raceId = race\_schedule.raceId

GROUP BY

driver\_rankings.driverId,

race\_schedule.year) AS driver\_season\_points

JOIN race\_schedule ON driver\_season\_points.year = race\_schedule.year

GROUP BY

decade

ORDER BY

decade;

**Q6. Who are the top 10 drivers with the most race wins, and what percentage of their total races did they win?**

SELECT

dd.forename,

dd.surname,

SUM(dr.wins) AS total\_wins,

COUNT(DISTINCT dr.raceId) AS total\_races,

ROUND(SUM(dr.wins) / COUNT(DISTINCT dr.raceId) \* 100, 1) AS win\_percentage

FROM

driver\_rankings dr

JOIN driver\_details dd ON dr.driverId = dd.driverId

GROUP BY

dr.driverId, dd.forename, dd.surname

ORDER BY

total\_wins DESC

LIMIT 10;

**Q7. Which constructors have dominated different eras (e.g., 1950s, 1960s, etc.) based on the number of constructor championships won?**

SELECT dd.forename, dd.surname, dd.nationality, dr.position, dr.wins, rs.year

FROM driver\_rankings AS dr

JOIN driver\_details AS dd

ON

dr.driverId = dd.driverId

JOIN race\_schedule AS rs

ON

rs.raceId = dr.raceId

GROUP BY

dd.forename, dd.surname, dd.nationality, dr.position, dr.wins, rs.year

ORDER BY rs.year desc;

Q8. Analyze the impact of regulation changes on team performance by comparing the points distribution before and after major rule changes.

SELECT td.constructorId, td.name, srr.position, srr.points, dd.forename, dd.surname,rs.year

FROM team\_details AS td

JOIN sprint\_race\_results AS srr

ON

td.constructorId = srr.constructorId

JOIN driver\_details AS dd

ON

srr.driverId = dd.driverId

JOIN race\_schedule AS rs

ON

srr.raceId = rs.raceId

GROUP BY

td.constructorId, td.name, srr.position, srr.points, dd.forename, dd.surname,rs.year

ORDER BY td.constructorId ASC;