

Mini Project Report

of

Database Systems Lab (CSE 2262)

Race Data Management System

SUBMITTED

BY

Tushar Pathak 230905396, CSE-B, 48

Riddhima Roy, 230905564, CSE-B, 65

Department of Computer Science and Engineering

**Manipal Institute of Technology, Manipal.**

**April 2024**

**TABLE OF CONTENTS**

* **Abstract**
* **Problem Statement**
* **ER Diagram and Relational Tables**
* **DDL Commands**
* **List of SQL Queries**
* **Procedures/ Triggers**
* **UI Design**
* **References**

**Abstract:**

The Formula 1 Database Management System is a comprehensive solution for organizing and managing all data related to Formula 1 racing. It covers everything from drivers, teams, circuits, and races to entire championship seasons. By maintaining clear connections between this information, the system makes it easy to analyze stats, track performance trends, and retrieve historical data.

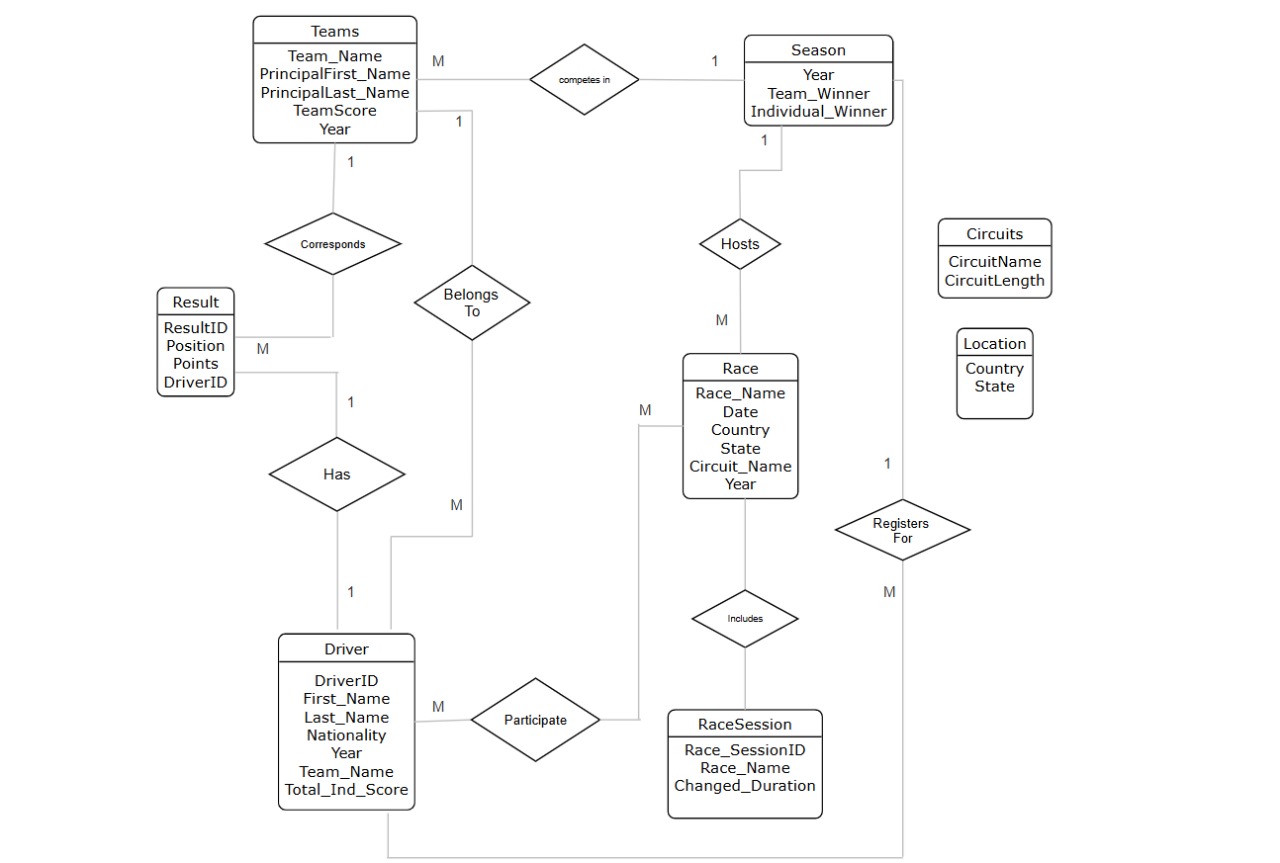
**Problem Statement:**

What problem are we solving?

Formula 1 is packed with data—drivers, teams, circuits, races, championships, and so much more. Managing this data manually or in a disorganized system is not only time-consuming but also prone to errors.

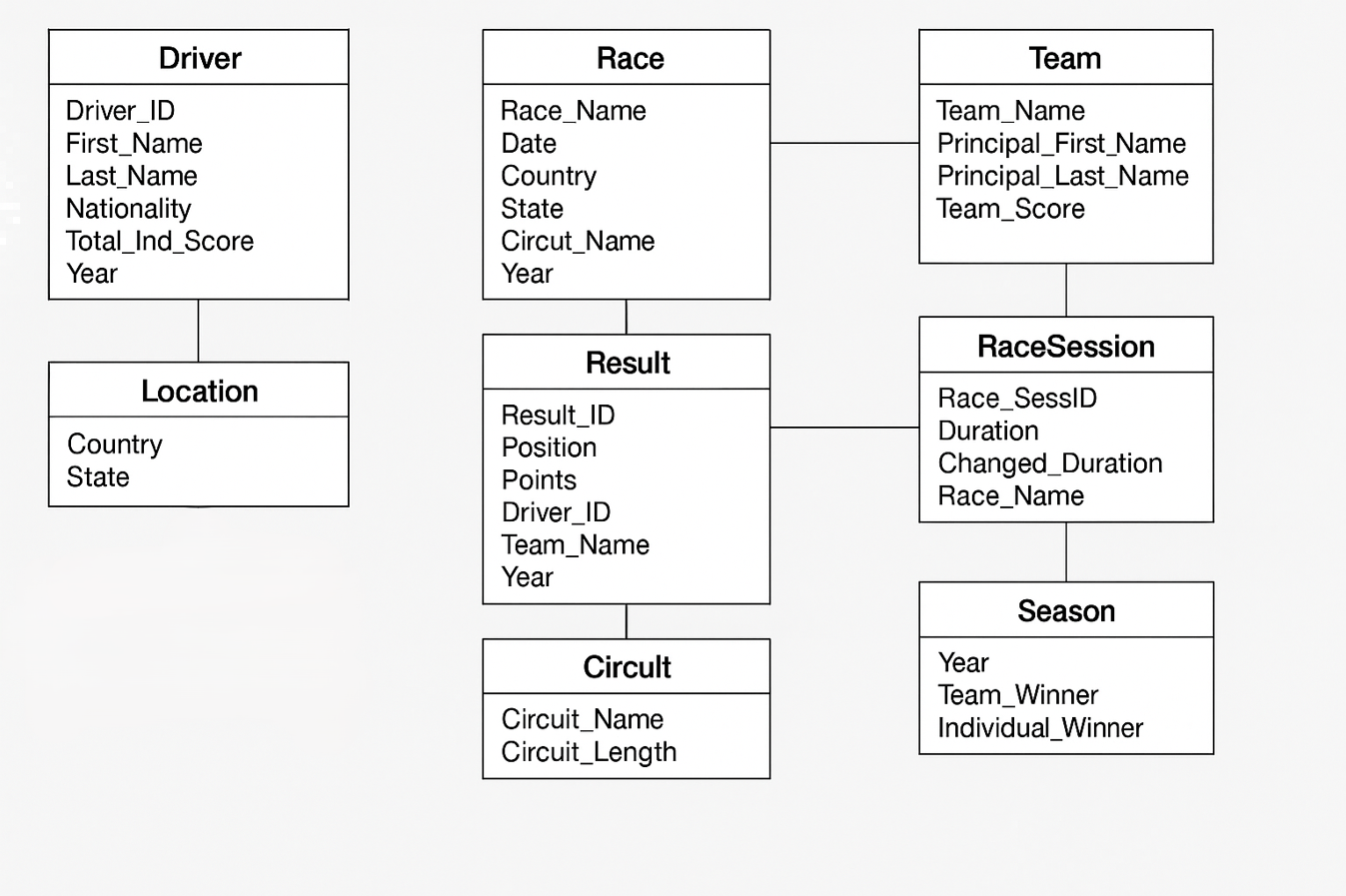
We aim to solve this by building a well-structured database that keeps everything organized, connected, and easy to access. Here's what we’ll focus on:

1. Drivers: Storing details like their names, nationalities, debut years, retirement status, and team history.
2. Teams: Capturing team names, principals, active years, and base locations.
3. Seasons and Championships: Tracking each season’s race schedule, championship winners, and team performances.
4. Races and Circuits: Keeping detailed records of races (dates, circuits, winners, lap records) and circuit information (location, length, fastest laps).
5. Race Results: Logging how drivers and teams performed in each race—positions, points, and fastest laps.
6. Historical Analysis: Making it easy to pull stats like top-performing drivers, team rankings, and season trends.

**ER Diagram**

**Relational Tables**

* Driver (**Driver\_ID**, First\_Name, Last\_Name, Nationality, Total\_Ind\_Score, **Year**, **Team\_Name**)
* Race (**Race\_Name**, Date, **Country**, **State**, **Circuit\_Name**, **Year**)
* Result (**Result\_ID**, Position, Points, **Driver\_ID**, **Team\_Name**, **Year**)
* Team (**Team\_Name**, Principal\_First\_Name, Principal\_Last\_Name, Team\_Score, **Year**)
* Circuit (**Circuit\_Name**, Circuit\_Length)
* Location (**Country**, **State**)
* RaceSession (**Race\_SessionID**, Duration, Changed\_Duration, **Race\_Name**)
* Season (**Year**, Team\_Winner, Individual\_Winner)

**Schema Diagram  
  
  
  
  
DDL Commands:**CREATE TABLE Season (

Year INT PRIMARY KEY,

Team\_Winner VARCHAR(255),

Individual\_Winner VARCHAR(255)

);

CREATE TABLE Location (

Country VARCHAR(255),

State VARCHAR(255),

PRIMARY KEY (Country, State)

);

CREATE TABLE Circuit (

Circuit\_Name VARCHAR(255) PRIMARY KEY,

Circuit\_Length DECIMAL(10, 2)

);

CREATE TABLE Team (

Team\_Name VARCHAR(255),

Principal\_First\_Name VARCHAR(255),

Principal\_Last\_Name VARCHAR(255),

Team\_Score INT,

Year INT,

PRIMARY KEY (Team\_Name, Year),

FOREIGN KEY (Year) REFERENCES Season (Year)

);

CREATE TABLE Driver (

Driver\_ID INT NOT NULL,

Nationality VARCHAR(255),

Last\_Name VARCHAR(255),

First\_Name VARCHAR(255),

Total\_Ind\_Score INT,

Year INT NOT NULL,

Team\_Name VARCHAR(255),

PRIMARY KEY (Driver\_ID, Year),

FOREIGN KEY (Year) REFERENCES Season (Year),

FOREIGN KEY (Team\_Name, Year) REFERENCES Team (Team\_Name, Year)

);

CREATE TABLE Race (

Race\_Name VARCHAR(255) PRIMARY KEY,

Race\_Date DATE,

Country VARCHAR(255),

State VARCHAR(255),

Circuit\_Name VARCHAR(255),

Year INT,

FOREIGN KEY (Circuit\_Name) REFERENCES Circuit (Circuit\_Name),

FOREIGN KEY (Year) REFERENCES Season (Year),

FOREIGN KEY (Country, State) REFERENCES Location (Country, State)

);

CREATE TABLE Result (

Result\_ID VARCHAR(255) PRIMARY KEY,

Position INT NOT NULL,

Points INT,

Driver\_ID INT NOT NULL,

Team\_Name VARCHAR(255),

Year INT NOT NULL,

FOREIGN KEY (Year) REFERENCES Season (Year),

FOREIGN KEY (Driver\_ID, Year) REFERENCES Driver (Driver\_ID, Year),

FOREIGN KEY (Team\_Name, Year) REFERENCES Team (Team\_Name, Year)

);

CREATE TABLE RaceSession (

Race\_SessionID VARCHAR(255) PRIMARY KEY,

Duration INT,

Changed\_Duration INT,

Race\_Name VARCHAR(255),

FOREIGN KEY (Race\_Name) REFERENCES Race (Race\_Name)

);

CREATE TABLE Audit\_Log (

Audit\_ID VARCHAR2(36),

Action\_Type VARCHAR2(10),

Table\_Name VARCHAR2(50),

Record\_ID VARCHAR2(100),

Action\_Details VARCHAR2(1000),

Action\_By VARCHAR2(50),

Action\_Date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (Audit\_ID)

); **Queries:  
  
Get Team Standings**:SELECT t.Team\_Name, t.Team\_Score, t.Principal\_First\_Name || ' ' || t.Principal\_Last\_Name as Principal, COUNT(DISTINCT d.Driver\_ID) AS Drivers\_Count

    FROM Team t

    LEFT JOIN Driver d ON t.Team\_Name = d.Team\_Name AND t.Year = d.Year

    WHERE t.Year = :year

    GROUP BY t.Team\_Name, t.Team\_Score, t.Principal\_First\_Name, t.Principal\_Last\_Name

    ORDER BY t.Team\_Score DESC

**Get Driver Standings**:SELECT d.Driver\_ID, d.First\_Name || ' ' || d.Last\_Name AS Driver\_Name,

           d.Total\_Ind\_Score, d.Team\_Name, d.Nationality

    FROM Driver d

    WHERE d.Year = :year

    ORDER BY d.Total\_Ind\_Score DESC  
  
 **Race Schedule**:

 SELECT r.Race\_Name, TO\_CHAR(r.Race\_Date, 'YYYY-MM-DD') as Race\_Date,

           r.Country, r.State, r.Circuit\_Name,

           c.Circuit\_Length

    FROM Race r

    JOIN Circuit c ON r.Circuit\_Name = c.Circuit\_Name

    WHERE r.Year = :year

    ORDER BY r.Race\_Date

**Championship History**:

 SELECT s.Year, s.Team\_Winner, s.Individual\_Winner,

           (SELECT MAX(Team\_Score) FROM Team WHERE Year = s.Year) AS Winning\_Team\_Score,

           (SELECT MAX(Total\_Ind\_Score) FROM Driver WHERE Year = s.Year) AS Winning\_Driver\_Score

    FROM Season s

    ORDER BY s.Year DESC

**Top Teams**:

SELECT t.Team\_Name, t.Team\_Score

        FROM Team t

        WHERE t.Year = {year}

        ORDER BY t.Team\_Score DESC

FETCH FIRST 5 ROWS ONLY

**Top Drivers**:

SELECT d.First\_Name || ' ' || d.Last\_Name as Driver\_Name, d.Total\_Ind\_Score

        FROM Driver d

        WHERE d.Year = {year}

        ORDER BY d.Total\_Ind\_Score DESC

        FETCH FIRST 5 ROWS ONLY

**Show Multiple Championships**:

SELECT s.Individual\_Winner, COUNT(\*) as championships

        FROM Season s

        GROUP BY s.Individual\_Winner

        HAVING COUNT(\*) > 1

        ORDER BY championships DESC

**Popular Circuits**:

SELECT c.Circuit\_Name, COUNT(r.Race\_Name) as race\_count

        FROM Circuit c

        JOIN Race r ON r.Circuit\_Name = c.Circuit\_Name

        WHERE r.Year = {year}

        GROUP BY c.Circuit\_Name

        ORDER BY race\_count DESC

        FETCH FIRST 5 ROWS ONLY

**Average Team Scores**:

SELECT t.Team\_Name, AVG(t.Team\_Score) as avg\_score

        FROM Team t

        WHERE t.Year = {year}

        GROUP BY t.Team\_Name

        ORDER BY avg\_score DESC

**Driver Improvements**:

SELECT d.First\_Name || ' ' || d.Last\_Name as Driver\_Name,

               (d.Total\_Ind\_Score - LAG(d.Total\_Ind\_Score) OVER (PARTITION BY d.First\_Name, d.Last\_Name ORDER BY d.Year)) as score\_diff

        FROM Driver d

        WHERE d.Year IN (SELECT Year FROM Season)

        ORDER BY score\_diff DESC NULLS LAST

        FETCH FIRST 5 ROWS ONLY

**Longest Race Sessions**:

SELECT r.Race\_Name, MAX(rs.Duration) as max\_duration

        FROM Race r

        JOIN RaceSession rs ON rs.Race\_Name = r.Race\_Name

        WHERE r.Year = {year}

        GROUP BY r.Race\_Name

        ORDER BY max\_duration DESC

        FETCH FIRST 5 ROWS ONLY

**Nationality Count**:

SELECT d.Nationality, COUNT(\*) as driver\_count

FROM Driver d

WHERE d.Year = {year}

GROUP BY d.Nationality

ORDER BY driver\_count DESC

**PL/SQL Queries**

**Find Driver Positions:**

CREATE OR REPLACE FUNCTION get\_driver\_position(

            p\_driver\_id IN NUMBER,

            p\_race\_name IN VARCHAR2

        ) RETURN NUMBER IS

            v\_position NUMBER;

        BEGIN

            SELECT Position INTO v\_position

            FROM Result

            WHERE Driver\_ID = p\_driver\_id

            AND SUBSTR(Result\_ID, 1, INSTR(Result\_ID, '-') - 1) = p\_race\_name;

            RETURN v\_position;

        EXCEPTION

            WHEN NO\_DATA\_FOUND THEN

                RETURN NULL;

            WHEN OTHERS THEN

                RETURN -1;

        END;

**Update Team Score**:

CREATE OR REPLACE PROCEDURE update\_team\_score\_proc (

p\_team\_name IN VARCHAR2,

p\_year IN NUMBER,

p\_additional\_points IN NUMBER

)

AS

BEGIN

UPDATE Team

SET Team\_Score = NVL(Team\_Score, 0) + p\_additional\_points

WHERE Team\_Name = p\_team\_name AND Year = p\_year;

IF SQL%ROWCOUNT = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('No team found with name "' || p\_team\_name || '" for year ' || p\_year);

ELSE

DBMS\_OUTPUT.PUT\_LINE('Team score updated successfully.');

END IF;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

ROLLBACK;

END;

**Race Session Count**:

CREATE OR REPLACE FUNCTION get\_race\_session\_count(

                p\_race\_name IN VARCHAR2

            ) RETURN NUMBER IS

                v\_count NUMBER;

            BEGIN

                SELECT COUNT(\*) INTO v\_count

                FROM RaceSession

                WHERE RacexEN NO\_DATA\_FOUND THEN

                    RETURN 0;

            END;

**Audit Log**:

CREATE OR REPLACE TRIGGER audit\_trigger

    AFTER INSERT OR UPDATE OR DELETE ON Team

    FOR EACH ROW

    DECLARE

        v\_action VARCHAR2(10);

        v\_record\_id VARCHAR2(100);

        v\_details VARCHAR2(1000);

    BEGIN

        IF INSERTING THEN

            v\_action := 'INSERT';

            v\_record\_id := :NEW.Team\_Name || '-' || :NEW.Year;

            v\_details := 'Team: ' || :NEW.Team\_Name || ', Score: ' || :NEW.Team\_Score ||

                        ', Principal: ' || :NEW.Principal\_First\_Name || ' ' || :NEW.Principal\_Last\_Name;

        ELSIF UPDATING THEN

            v\_action := 'UPDATE';

            v\_record\_id := :NEW.Team\_Name || '-' || :NEW.Year;

            v\_details := 'Old Score: ' || :OLD.Team\_Score || ', New Score: ' || :NEW.Team\_Score ||

                        ', Old Principal: ' || :OLD.Principal\_First\_Name || ' ' || :OLD.Principal\_Last\_Name ||

                        ', New Principal: ' || :NEW.Principal\_First\_Name || ' ' || :NEW.Principal\_Last\_Name;

        ELSE

            v\_action := 'DELETE';

            v\_record\_id := :OLD.Team\_Name || '-' || :OLD.Year;

            v\_details := 'Team: ' || :OLD.Team\_Name || ', Score: ' || :OLD.Team\_Score;

        END IF;

        INSERT INTO Audit\_Log (

            Audit\_ID,

            Action\_Type,

            Table\_Name,

            Record\_ID,

            Action\_Details,

            Action\_By

        ) VALUES (

            SYS\_GUID(),

            v\_action,

            'Team',

            v\_record\_id,

            v\_details,

            USER

        );

    END;

**UI Design**

A screenshot of a computer

AI-generated content may be incorrect.

Fig 1. Login Page

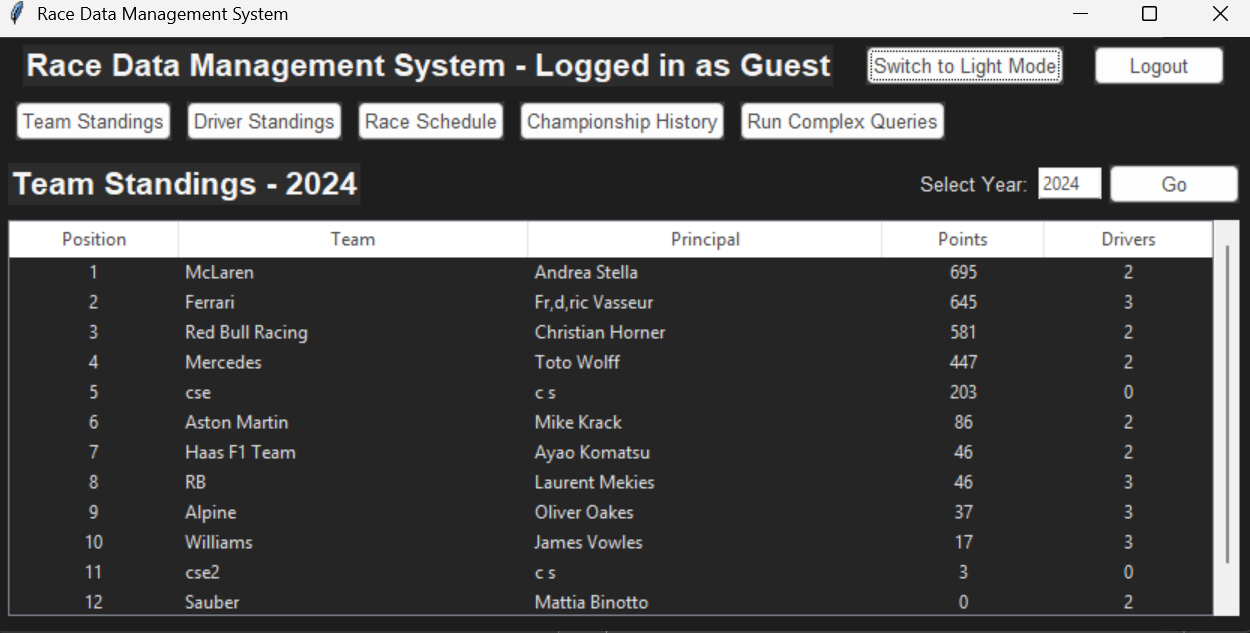


Fig 2. Guest Interface – Team Standings Tab

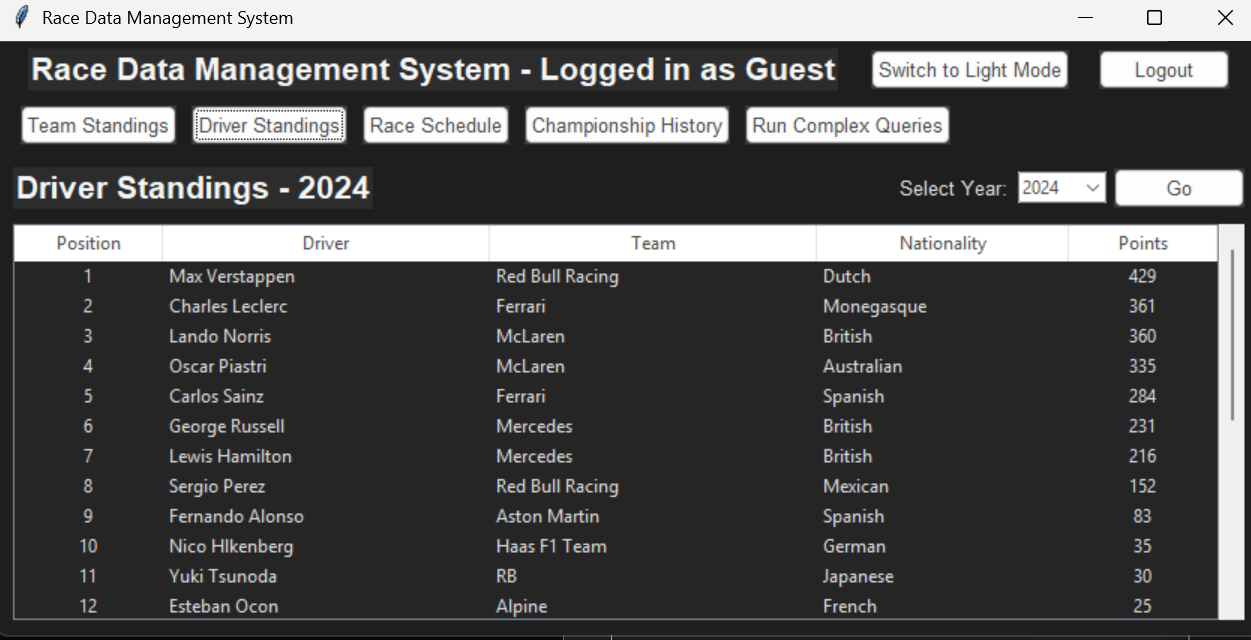


Fig 3. Driver Standings Tab

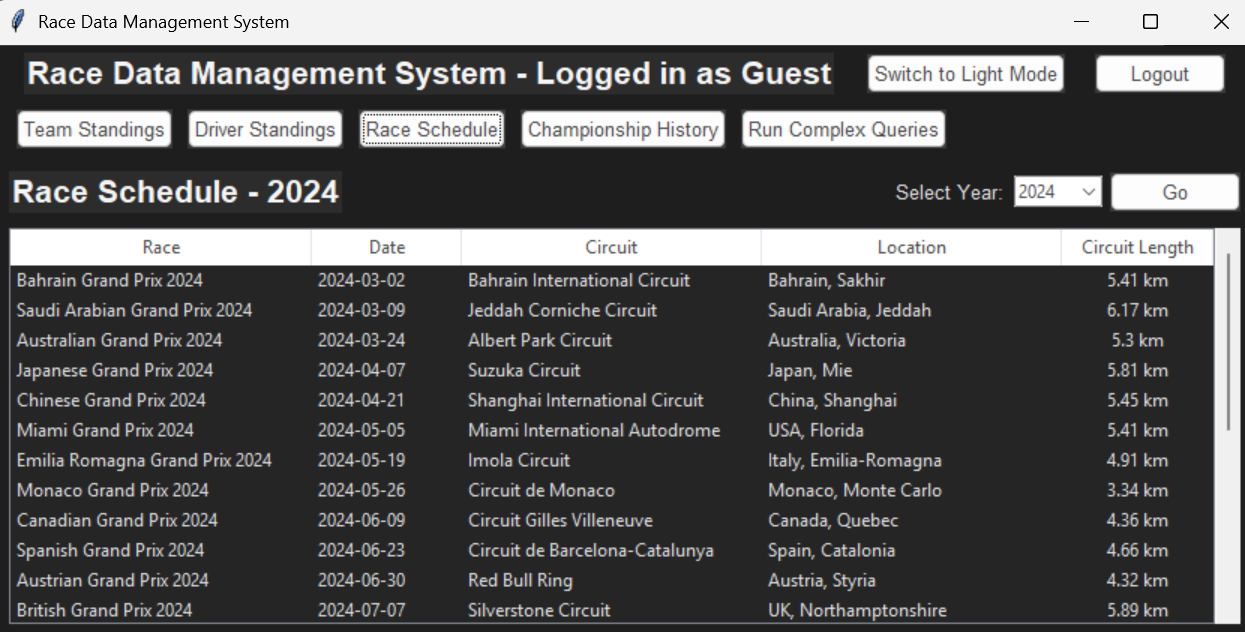


Fig 4. Race Schedule Tab

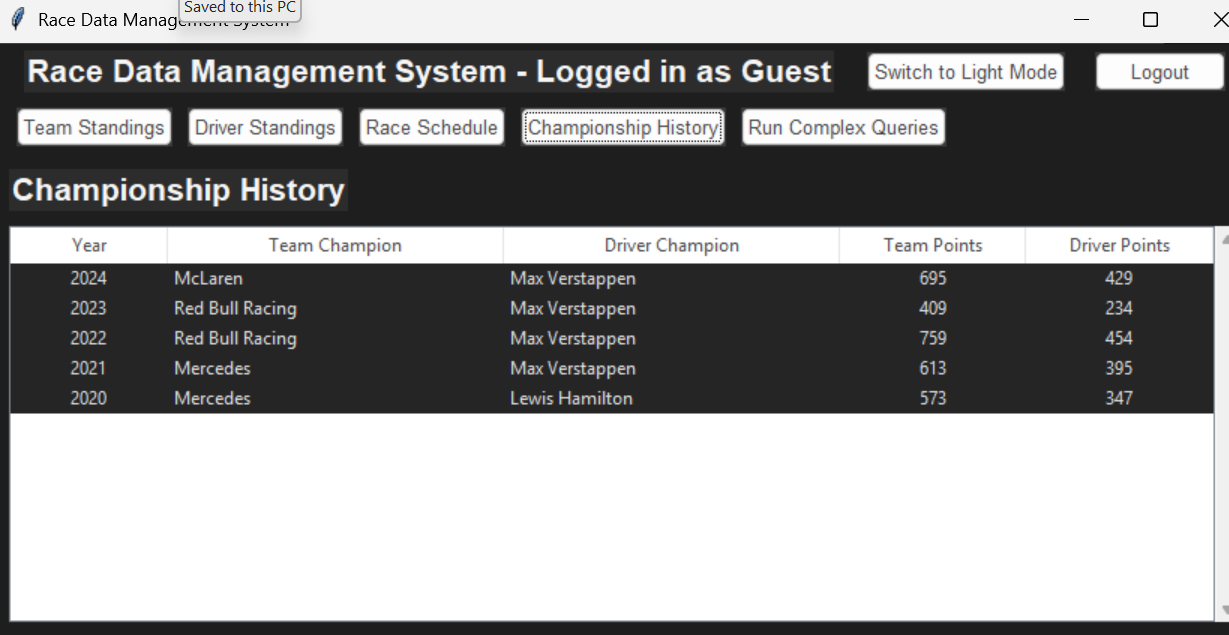


Fig 5. Championship History Tab

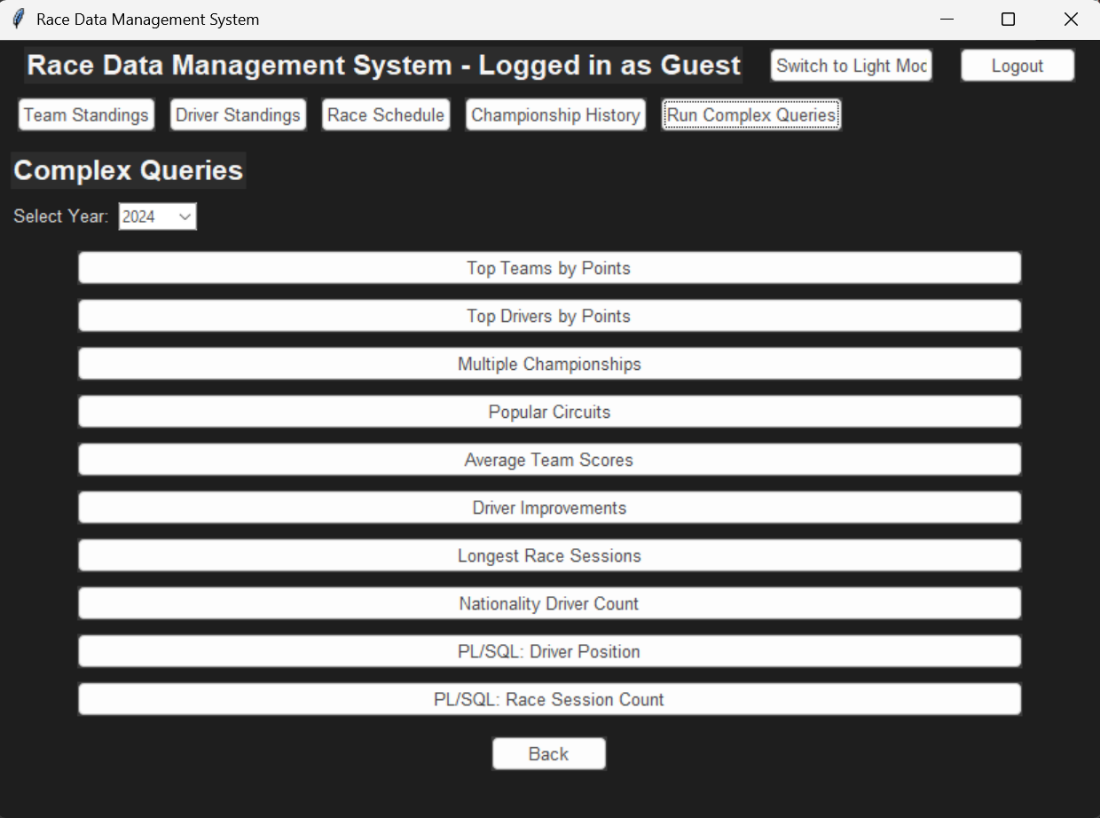


Fig 6. Complex Queries Tab

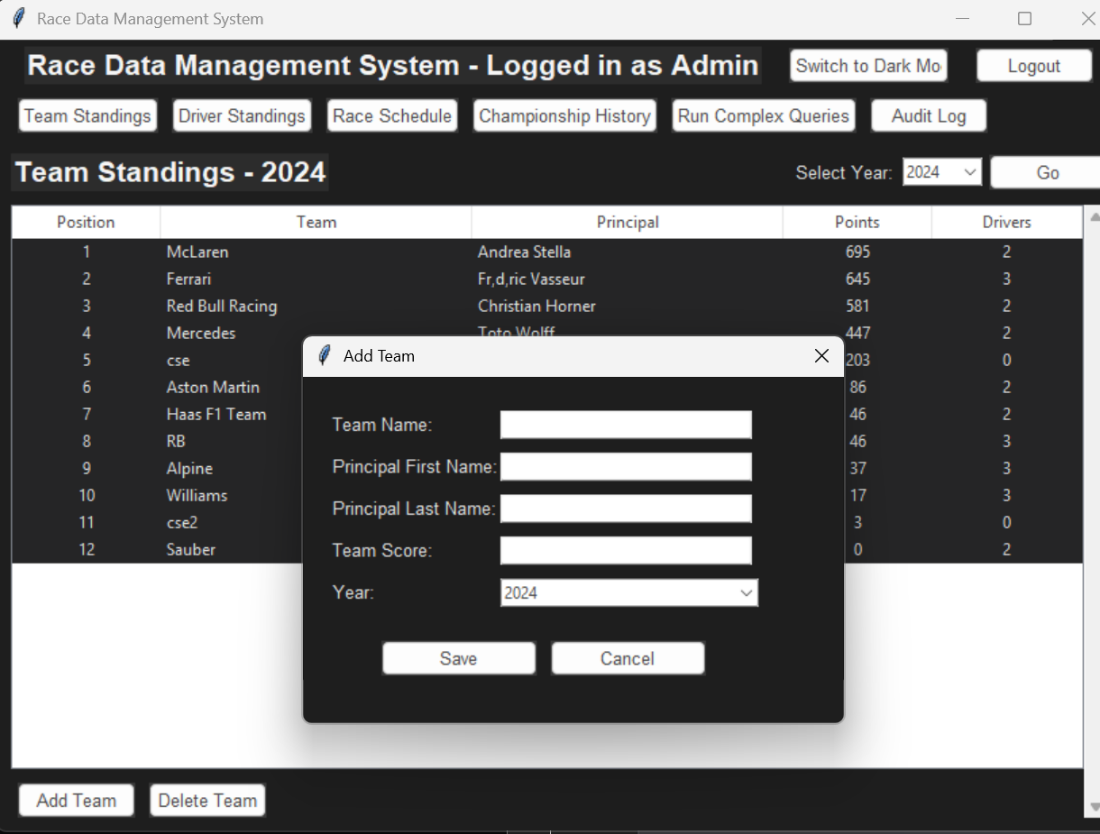


Fig 7. Add Team

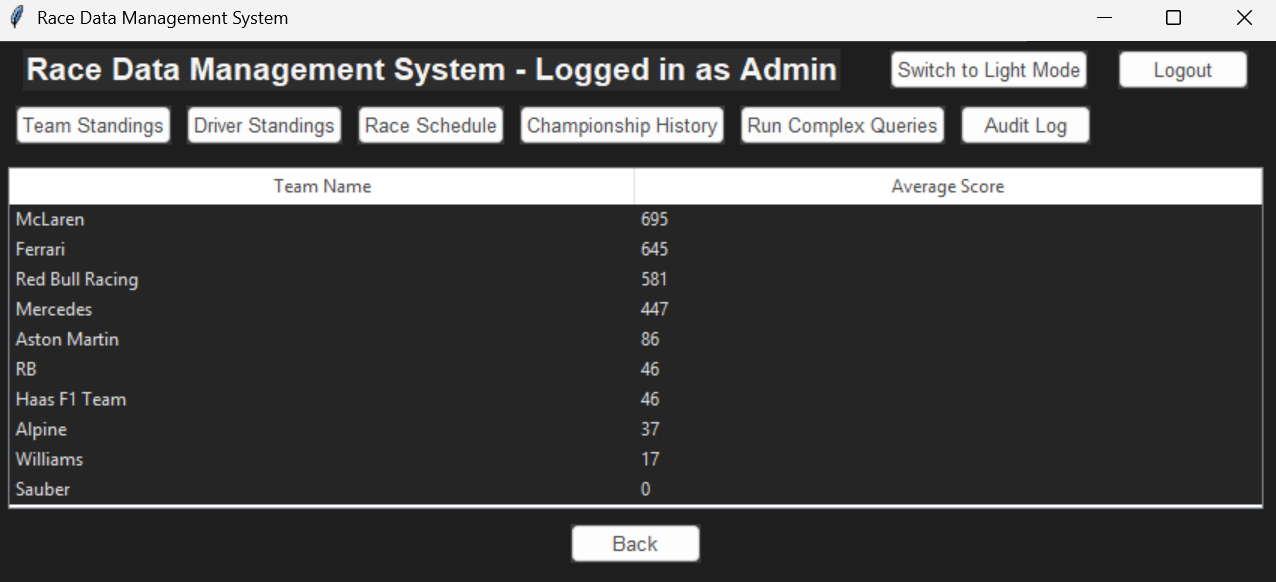


Fig 8. Average Team Scores Query Result

A screen shot of a computer

AI-generated content may be incorrect.

Fig 9. Nationality Count Query Result

A screenshot of a computer

AI-generated content may be incorrect.

Fig 10. Find Driver Position PL/SQL Query

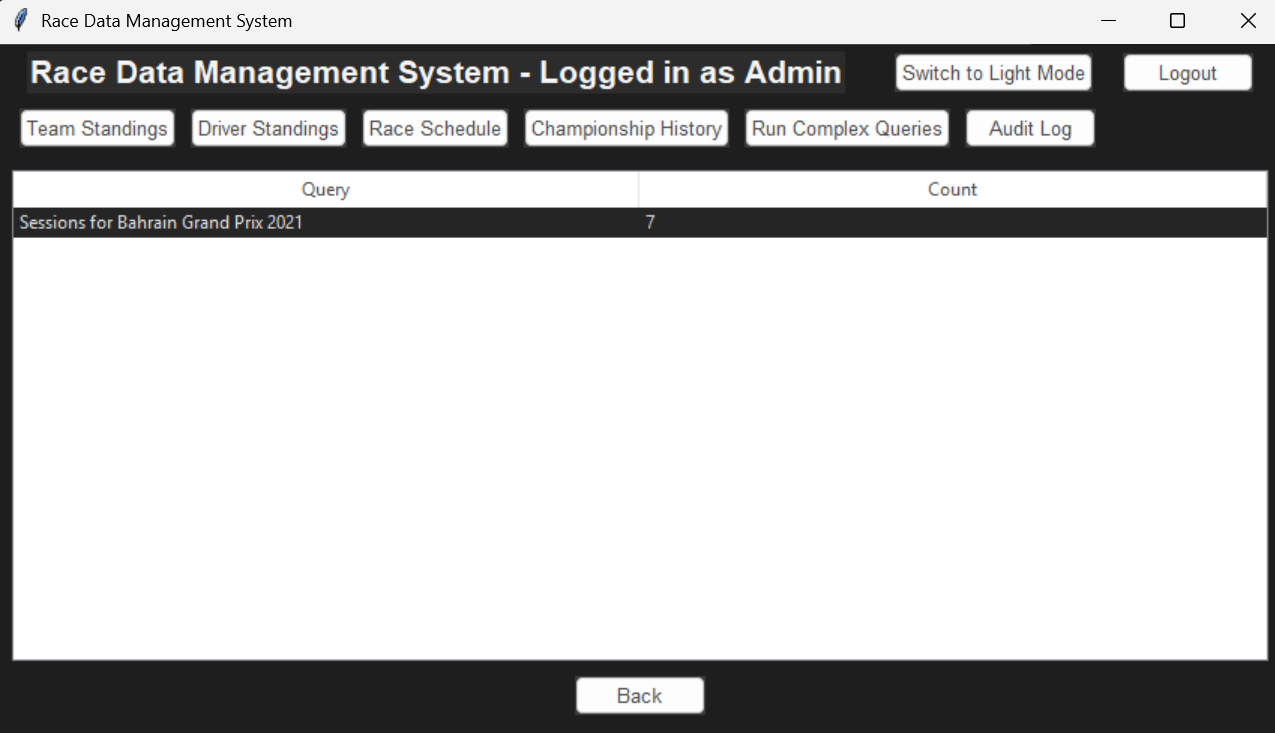


Fig 11. Find Race Session Count PL/SQL Query

**References**:

1. **Oracle Database 21c Documentation**  
   <https://docs.oracle.com/en/database/oracle/oracle-database/21/>
2. **Oracle PL/SQL Programming**  
   <https://www.oreilly.com/library/view/oracle-plsql-programming/9781449324445/>
3. **Expert Oracle Database Architecture**  
   <https://link.springer.com/book/10.1007/978-1-4302-6299-2>
4. **Python and Tkinter Programming**  
   <https://www.manning.com/books/python-and-tkinter-programming>
5. **Tkinter — Python Interface**   
   <https://docs.python.org/3/library/tkinter.html>
6. **Python CX\_Oracle Documentation**  
   <https://cx-oracle.readthedocs.io/en/latest/>
7. **Python Programming on Win32**  
   <https://www.oreilly.com/library/view/python-programming-on/1565926218/>
8. **Flask Web Development**  
   <https://www.oreilly.com/library/view/flask-web-development/9781491991725/>
9. **Flask Documentation**  
   <https://flask.palletsprojects.com/en/2.3.x/>
10. **Formula 1 Regulations**  
    <https://www.fia.com/en/regulations>
11. **The Art of the Formula 1 Race Car**  
    <https://www.quartoknows.com/books/9780760368374/The-Art-of-the-Formula-1-Race-Car.html>