Database Management Systems Project-

StatChicane: An F1 Stats Agent

Members:

Sheerin Fatima Aijaz (RA2311030010063)
Sarthak Suwan (RA2311030010094)

Department:

Networking and Communications

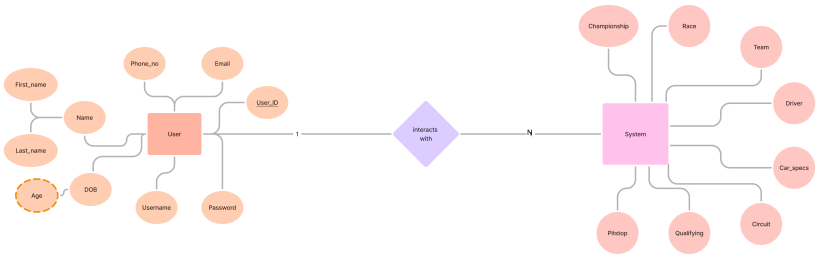
Section:

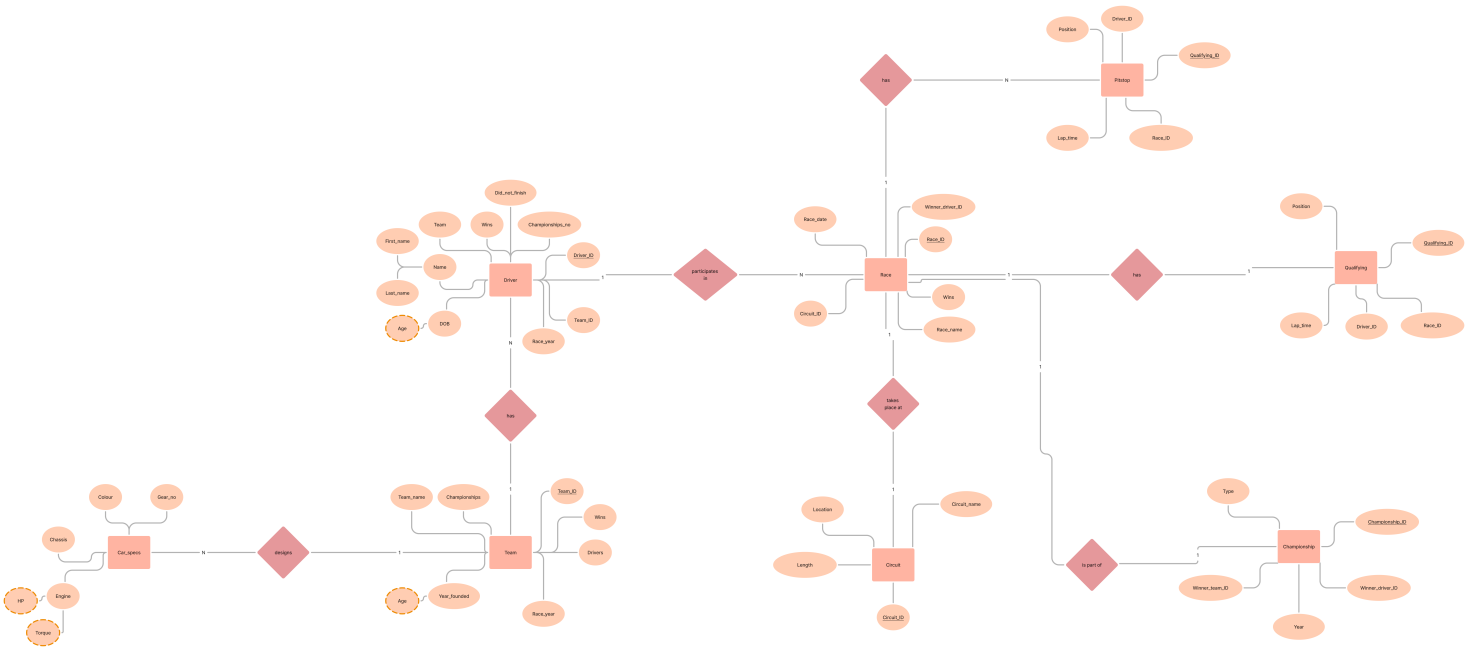
Y1 section

Primary Objective / Problem Statement:

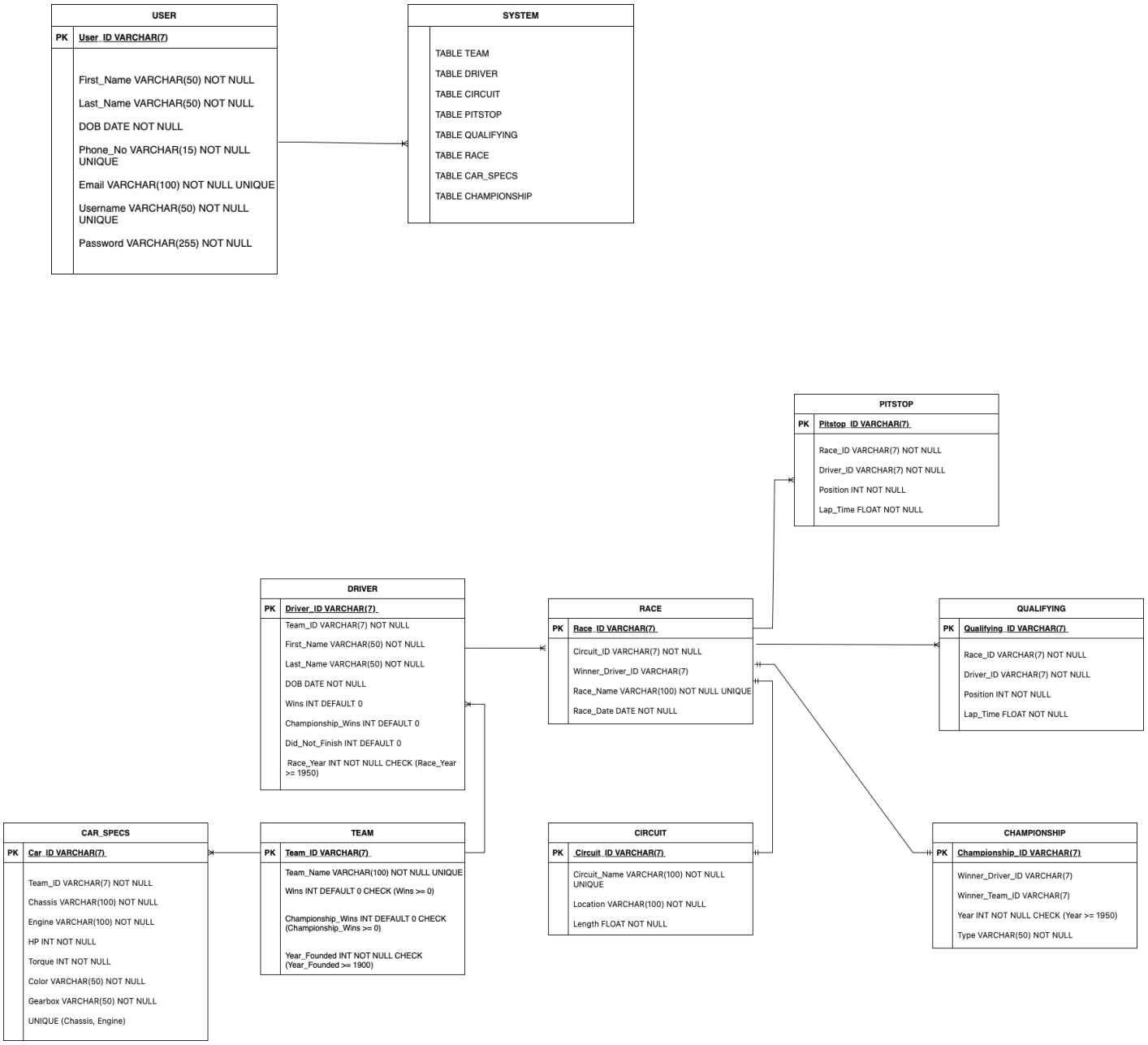
To create a user-friendly F1 statistics prediction and querying system using natural language processing (NLP) and a database management system, as F1 commentators and fans often seek specific, real-time statistics about races, drivers, and teams. Manually querying databases for such data is time-consuming and requires technical expertise.

ER diagram :-


First_name
Name
Last_name
Age
DOB
Phone_no
Email
User_ID
User
Username
Password
1
Interacts with
N
System
Championship
Race
Team
Driver
Car_specs
Circuit
Qualifying
Pitstop


Position
Driver_ID
Qualifying_ID
Lap_time
Race_ID
Pitstop
has
N
1
Did_not_finish
Team
Wins
Championships_no
First_name
Name
Last_name
Age
DOB
Driver_ID
Team_ID
Race_year
Driver
participates in
Race_date
Winner_driver_ID
Race_ID
Circuit_ID
Wins
Race_name
Race
has
Position
Qualifying_ID
Lap_time
Driver_ID
Race_ID
Qualifying
Colour
Gear_no
Chassis
HP
Engine
Torque
Car_specs
designs
Team_name
Championships
Team_ID
Wins
Drivers
Age
Year_founded
Team
Race_year
takes place at
Location
Circuit_name
Length
Circuit
Circuit_ID
is part of
Type
Championship_ID
Winner_team_ID
Winner_driver_ID
Year
Championship

SCHEMA :-


USER
PK
User_ID VARCHAR(7)
First_Name VARCHAR(50) NOT NULL
Last_Name VARCHAR(50) NOT NULL
DOB DATE NOT NULL
Phone_No VARCHAR(15) NOT NULL UNIQUE
Email VARCHAR(100) NOT NULL UNIQUE
Username VARCHAR(50) NOT NULL UNIQUE
Password VARCHAR(255) NOT NULL
SYSTEM
TABLE TEAM
TABLE DRIVER
TABLE CIRCUIT
TABLE PITSTOP
TABLE QUALIFYING
TABLE RACE
TABLE CAR_SPECS
TABLE CHAMPIONSHIP
PITSTOP
PK
Pitstop_ID VARCHAR(7)
Race_ID VARCHAR(7) NOT NULL
Driver_ID VARCHAR(7) NOT NULL
Position INT NOT NULL
Lap_Time FLOAT NOT NULL
DRIVER
PK
Driver_ID VARCHAR(7)
Team_ID VARCHAR(7) NOT NULL
First_Name VARCHAR(50) NOT NULL
Last_Name VARCHAR(50) NOT NULL
DOB DATE NOT NULL
Wins INT DEFAULT 0
Championship_Wins INT DEFAULT 0
Did_Not_Finish INT DEFAULT 0
Race_Year INT NOT NULL CHECK (Race_Year >= 1950)
RACE
PK
Race_ID VARCHAR(7)
Circuit_ID VARCHAR(7) NOT NULL
Winner_Driver_ID VARCHAR(7)
Race_Name VARCHAR(100) NOT NULL UNIQUE
Race_Date DATE NOT NULL
QUALIFYING
PK
Qualifying_ID VARCHAR(7)
Race_ID VARCHAR(7) NOT NULL
Driver_ID VARCHAR(7) NOT NULL
Position INT NOT NULL
Lap_Time FLOAT NOT NULL
CAR_SPECS
PK
Car_ID VARCHAR(7)
Team_ID VARCHAR(7) NOT NULL
Chassis VARCHAR(100) NOT NULL
Engine VARCHAR(100) NOT NULL
HP INT NOT NULL
Torque INT NOT NULL
Color VARCHAR(50) NOT NULL
Gearbox VARCHAR(50) NOT NULL
UNIQUE (Chassis, Engine)
TEAM
PK
Team_ID VARCHAR(7)
Team_Name VARCHAR(100) NOT NULL UNIQUE
Wins INT DEFAULT 0 CHECK (Wins >= 0)
Championship_Wins INT DEFAULT 0 CHECK (Championship_Wins >= 0)
Year_Founded INT NOT NULL CHECK (Year_Founded >= 1900)
CIRCUIT
PK
Circuit_ID VARCHAR(7)
Circuit_Name VARCHAR(100) NOT NULL UNIQUE
Location VARCHAR(100) NOT NULL
Length FLOAT NOT NULL
CHAMPIONSHIP
PK
Championship_ID VARCHAR(7)
Winner_Driver_ID VARCHAR(7)
Winner_Team_ID VARCHAR(7)
Year INT NOT NULL CHECK (Year >= 1950)
Type VARCHAR(50) NOT NULL

RELATIONAL TABLES :-

1. Team

Attribute	Type	Constraint
Team_ID	VARCHAR(7)	Primary Key
Team_Name	VARCHAR(100)	NOT NULL, UNIQUE
Wins	INT	DEFAULT 0, CHECK (Wins >= 0)
Championship_Wins	INT	DEFAULT 0, CHECK (Championship_Wins >= 0)
Year_Founded	INT	NOT NULL, CHECK (Year_Founded >= 1900)

2. Driver

Attribute	Type	Constraint
Driver_ID	VARCHAR(7)	Primary Key
First_Name	VARCHAR(50)	NOT NULL
Last_Name	VARCHAR(50)	NOT NULL
DOB	DATE	NOT NULL
Wins	INT	DEFAULT 0
Championship_Wins	INT	DEFAULT 0
Did_Not_Finish	INT	DEFAULT 0
Team_ID	VARCHAR(7)	Foreign Key → Team.Team_ID, NOT NULL

Race_Year	INT	NOT NULL, CHECK (Race_Year >= 1950)

3. Car_Specs

Attribute	Type	Constraint
Car_ID	VARCHAR(7)	Primary Key
Chassis	VARCHAR(100)	NOT NULL
Engine	VARCHAR(100)	NOT NULL
HP	INT	NOT NULL
Torque	INT	NOT NULL
Color	VARCHAR(50)	NOT NULL
Gearbox	VARCHAR(50)	NOT NULL
Team_ID	VARCHAR(7)	Foreign Key → Team.Team_ID, NOT NULL

4. Circuit

Attribute	Type	Constraint
Circuit_ID	VARCHAR(7)	Primary Key
Circuit_Name	VARCHAR(100)	NOT NULL, UNIQUE
Location	VARCHAR(100)	NOT NULL
Length	FLOAT	NOT NULL

5. Championship

Attribute	Type	Constraint
Championship_ID	VARCHAR(7)	Primary Key
Year	INT	NOT NULL, CHECK (Year >= 1950)
Type	VARCHAR(50)	NOT NULL
Winner_Driver_ID	VARCHAR(7)	Foreign Key → Driver.Driver_ID
Winner_Team_ID	VARCHAR(7)	Foreign Key → Team.Team_ID

6. Race

Attribute	Type	Constraint
Race_ID	VARCHAR(7)	Primary Key
Race_Name	VARCHAR(100)	NOT NULL, UNIQUE
Race_Date	DATE	NOT NULL
Circuit_ID	VARCHAR(7)	Foreign Key → Circuit.Circuit_ID
Winner_Driver_ID	VARCHAR(7)	Foreign Key → Driver.Driver_ID

7. Pitstop

Attribute	Type	Constraint
Pitstop_ID	VARCHAR(7)	Primary Key
Race_ID VARCHAR(7)** FOREIGN KEY		

CODE :-

```
CREATE TABLE Team (
    Team_ID VARCHAR(7) PRIMARY KEY,
    Team_Name VARCHAR(100) NOT NULL UNIQUE,
    Wins INT DEFAULT 0 CHECK (Wins >= 0),
    Championship_Wins INT DEFAULT 0 CHECK (Championship_Wins >= 0),
    Year_Founded INT NOT NULL CHECK (Year_Founded >= 1900)
);

CREATE TABLE Driver (
    Driver_ID VARCHAR(7) PRIMARY KEY,
    First_Name VARCHAR(50) NOT NULL,
    Last_Name VARCHAR(50) NOT NULL,
    DOB DATE NOT NULL,
    Wins INT DEFAULT 0,
    Championship_Wins INT DEFAULT 0,
    Did_Not_Finish INT DEFAULT 0,
    Team_ID VARCHAR(7) NOT NULL,
    Race_Year INT NOT NULL CHECK (Race_Year >= 1950),
    FOREIGN KEY (Team_ID) REFERENCES Team(Team_ID)
);

CREATE TABLE Car_Specs (
    Car_ID VARCHAR(7) PRIMARY KEY,
    Chassis VARCHAR(100) NOT NULL,
    Engine VARCHAR(100) NOT NULL,
    HP INT NOT NULL,
    Torque INT NOT NULL,
    Color VARCHAR(50) NOT NULL,
    Gearbox VARCHAR(50) NOT NULL,
    Team_ID VARCHAR(7) NOT NULL,
    UNIQUE (Chassis, Engine),
    FOREIGN KEY (Team_ID) REFERENCES Team(Team_ID)
);
```

```
CREATE TABLE Circuit (
    Circuit_ID VARCHAR(7) PRIMARY KEY,
    Circuit_Name VARCHAR(100) NOT NULL UNIQUE,
    Location VARCHAR(100) NOT NULL,
    Length FLOAT NOT NULL
);

CREATE TABLE Championship (
    Championship_ID VARCHAR(7) PRIMARY KEY,
    Year INT NOT NULL CHECK (Year >= 1950),
    Type VARCHAR(50) NOT NULL,
    Winner_Driver_ID VARCHAR(7),
    Winner_Team_ID VARCHAR(7),
    FOREIGN KEY (Winner_Driver_ID) REFERENCES Driver(Driver_ID),
    FOREIGN KEY (Winner_Team_ID) REFERENCES Team(Team_ID)
);

CREATE TABLE Race (
    Race_ID VARCHAR(7) PRIMARY KEY,
    Race_Name VARCHAR(100) NOT NULL UNIQUE,
    Race_Date DATE NOT NULL,
    Circuit_ID VARCHAR(7) NOT NULL,
    Winner_Driver_ID VARCHAR(7),
    FOREIGN KEY (Circuit_ID) REFERENCES Circuit(Circuit_ID),
    FOREIGN KEY (Winner_Driver_ID) REFERENCES Driver(Driver_ID)
);

CREATE TABLE Pitstop (
    Pitstop_ID VARCHAR(7) PRIMARY KEY,
    Race_ID VARCHAR(7) NOT NULL,
    Driver_ID VARCHAR(7) NOT NULL,
    Position INT NOT NULL,
    Lap_Time FLOAT NOT NULL,
    FOREIGN KEY (Race_ID) REFERENCES Race(Race_ID),
    FOREIGN KEY (Driver_ID) REFERENCES Driver(Driver_ID)
);
```

```
CREATE TABLE Qualifying (
    Qualifying_ID VARCHAR(7) PRIMARY KEY,
    Race_ID VARCHAR(7) NOT NULL,
    Driver_ID VARCHAR(7) NOT NULL,
    Position INT NOT NULL,
    Lap_Time FLOAT NOT NULL,
    FOREIGN KEY (Race_ID) REFERENCES Race(Race_ID),
    FOREIGN KEY (Driver_ID) REFERENCES Driver(Driver_ID)
);

-- Now Adding the relationship table since we are using MANY to MANY Rellationship

-- Participates Relationship (Many-to-Many)
CREATE TABLE Participates (
    Driver_ID VARCHAR(7) NOT NULL,
    Race_ID VARCHAR(7) NOT NULL,
    PRIMARY KEY (Driver_ID, Race_ID),
    FOREIGN KEY (Driver_ID) REFERENCES Driver(Driver_ID),
    FOREIGN KEY (Race_ID) REFERENCES Race(Race_ID)
);

-- Takes Place At Relationship
CREATE TABLE Takes_Place_At (
    Race_ID VARCHAR(7) NOT NULL,
    Circuit_ID VARCHAR(7) NOT NULL,
    PRIMARY KEY (Race_ID, Circuit_ID),
    FOREIGN KEY (Race_ID) REFERENCES Race(Race_ID),
    FOREIGN KEY (Circuit_ID) REFERENCES Circuit(Circuit_ID)
);

-- Belongs To Relationship
CREATE TABLE Belongs_To (
    Car_ID VARCHAR(7) NOT NULL,
    Team_ID VARCHAR(7) NOT NULL,
    PRIMARY KEY (Car_ID, Team_ID),
```

```
    FOREIGN KEY (Car_ID) REFERENCES Car_Specs(Car_ID),
    FOREIGN KEY (Team_ID) REFERENCES Team(Team_ID)
);

-- Is Part Of Relationship
CREATE TABLE Is_Part_Of (
    Race_ID VARCHAR(7) NOT NULL,
    Championship_ID VARCHAR(7) NOT NULL,
    PRIMARY KEY (Race_ID, Championship_ID),
    FOREIGN KEY (Race_ID) REFERENCES Race(Race_ID),
    FOREIGN KEY (Championship_ID) REFERENCES Championship(Championship_ID)
);

-- The main user table
CREATE TABLE User (
    User_ID VARCHAR(7) PRIMARY KEY,
    First_Name VARCHAR(50) NOT NULL,
    Last_Name VARCHAR(50) NOT NULL,
    DOB DATE NOT NULL,
    Phone_No VARCHAR(15) NOT NULL UNIQUE,
    Email VARCHAR(100) NOT NULL UNIQUE,
    Username VARCHAR(50) NOT NULL UNIQUE,
    Password VARCHAR(255) NOT NULL  -- Store hashed passwords securely
)
```

Insertion

```
-- Inserting Values
INSERT INTO Team (Team_ID, Team_Name, Wins, Championship_Wins, Year_Founded)
VALUES
    ('T001', 'Oracle Red Bull Racing', 11, 5, 2005),
    ('T002', 'McLaren F1 Team', 4, 1, 1966),
    ('T003', 'Scuderia Ferrari', 3, 0, 1950),
```

```
    ('T004', 'Mercedes-AMG Petronas F1 Team', 2, 0, 2010);

   INSERT INTO Driver (Driver_ID, First_Name, Last_Name, DOB, Wins, Championship_Wins,
Did_Not_Finish, Team_ID, Race_Year)
VALUES
    ('D001', 'Max', 'Verstappen', '1997-09-30', 9, 4, 1, 'T001', 2024),
    ('D002', 'Lando', 'Norris', '1999-11-13', 4, 0, 0, 'T002', 2024),
    ('D003', 'Charles', 'Leclerc', '1997-10-16', 3, 0, 0, 'T003', 2024),
    ('D004', 'Lewis', 'Hamilton', '1985-01-07', 2, 7, 1, 'T004', 2024);

    INSERT INTO Circuit (Circuit_ID, Circuit_Name, Location, Length)
VALUES
    ('C001', 'Silverstone Circuit', 'Silverstone, UK', 5.891),
    ('C002', 'Autódromo José Carlos Pace', 'São Paulo, Brazil', 4.309);

    INSERT INTO Race (Race_ID, Race_Name, Race_Date, Circuit_ID, Winner_Driver_ID)
VALUES
    ('R001', 'British Grand Prix', '2024-07-07', 'C001', 'D002'),
    ('R002', 'São Paulo Grand Prix', '2024-11-17', 'C002', 'D001');

    -- links races to circuits
    INSERT INTO Takes_Place_At (Race_ID, Circuit_ID)
VALUES
    ('R001', 'C001'),
    ('R002', 'C002');

    -- refisters driver participation into races
    INSERT INTO Participates (Driver_ID, Race_ID)
VALUES
    ('D001', 'R001'),
    ('D002', 'R001'),
    ('D003', 'R001'),
    ('D004', 'R001'),
    ('D001', 'R002'),
    ('D002', 'R002'),
    ('D003', 'R002'),
```

```
    ('D004', 'R002');

    INSERT INTO Car_Specs (Car_ID, Chassis, Engine, HP, Torque, Color, Gearbox, Team_ID)
VALUES
    ('C001', 'RB20', 'Honda RA621H', 1000, 500, 'Red', '8-speed', 'T001'),
    ('C002', 'MCL38', 'Mercedes-AMG F1 M14', 980, 490, 'Papaya Orange', '8-speed', 'T002'),
    ('C003', 'SF24', 'Ferrari 066/10', 970, 485, 'Red', '8-speed', 'T003'),
    ('C004', 'W15', 'Mercedes-AMG F1 M14', 980, 490, 'Silver', '8-speed', 'T004');

    -- assigining cars to teams
    INSERT INTO Belongs_To (Car_ID, Team_ID)
VALUES
    ('C001', 'T001'),
    ('C002', 'T002'),
    ('C003', 'T003'),
    ('C004', 'T004');

    INSERT INTO Qualifying (Qualifying_ID, Race_ID, Driver_ID, Position, Lap_Time)
VALUES
    ('Q001', 'R001', 'D002', 1, 90.123),
    ('Q002', 'R001', 'D001', 2, 90.456),
    ('Q003', 'R001', 'D003', 3, 90.789),
    ('Q004', 'R001', 'D004', 4, 91.012),
    ('Q005', 'R002', 'D001', 1, 88.123),
    ('Q006', 'R002', 'D003', 2, 88.456),
    ('Q007', 'R002', 'D002', 3, 88.789),
    ('Q008', 'R002', 'D004', 4, 89.012);

    INSERT INTO Pitstop (Pitstop_ID, Race_ID, Driver_ID, Position, Lap_Time)
VALUES
    ('P001', 'R001', 'D001', 1, 21.5),
    ('P002', 'R001', 'D002', 1, 21.3),
    ('P003', 'R001', 'D003', 1, 21.7),
    ('P004', 'R001', 'D004', 1, 21.6),
    ('P005', 'R002', 'D001', 1, 22.0),
    ('P006', 'R002', 'D002', 1, 21.8),
```

```
    ('P007', 'R002', 'D003', 1, 22.1),
    ('P008', 'R002', 'D004', 1, 21.9);
-- user table input
    INSERT INTO User (User_ID, First_Name, Last_Name, DOB, Phone_No, Email, Username,
Password)
VALUES
    ('U001', 'Sarthak', 'Suwan', '2004-10-30', '9876543210', 'sarthak@example.com', 'sarthak_20',
'hashed_password_here');

    -- views

    -- View for Team table
CREATE VIEW Team_View AS
SELECT Team_ID, Team_Name, Wins, Championship_Wins, Year_Founded
FROM Team;

-- View for Driver table
CREATE VIEW Driver_View AS
SELECT Driver_ID, First_Name, Last_Name, DOB, Wins, Championship_Wins, Did_Not_Finish,
Team_ID, Race_Year
FROM Driver;

-- View for Car_Specs table
CREATE VIEW Car_Specs_View AS
SELECT Car_ID, Chassis, Engine, HP, Torque, Color, Gearbox, Team_ID
FROM Car_Specs;

-- View for Circuit table
CREATE VIEW Circuit_View AS
SELECT Circuit_ID, Circuit_Name, Location, Length
FROM Circuit;

-- View for Championship table
CREATE VIEW Championship_View AS
SELECT Championship_ID, Year, Type, Winner_Driver_ID, Winner_Team_ID
FROM Championship;
```

```
-- View for Race table
CREATE VIEW Race_View AS
SELECT Race_ID, Race_Name, Race_Date, Circuit_ID, Winner_Driver_ID
FROM Race;

-- View for Pitstop table
CREATE VIEW Pitstop_View AS
SELECT Pitstop_ID, Race_ID, Driver_ID, Position, Lap_Time
FROM Pitstop;

-- View for Qualifying table
CREATE VIEW Qualifying_View AS
SELECT Qualifying_ID, Race_ID, Driver_ID, Position, Lap_Time
FROM Qualifying;

-- View for Participates (Many-to-Many relationship between Driver and Race)
CREATE VIEW Participates_View AS
SELECT Driver_ID, Race_ID
FROM Participates;

-- View for Takes_Place_At (Many-to-Many relationship between Race and Circuit)
CREATE VIEW Takes_Place_At_View AS
SELECT Race_ID, Circuit_ID
FROM Takes_Place_At;

-- View for Belongs_To (Many-to-Many relationship between Car and Team)
CREATE VIEW Belongs_To_View AS
SELECT Car_ID, Team_ID
FROM Belongs_To;

-- View for Is_Part_Of (Many-to-Many relationship between Race and Championship)
CREATE VIEW Is_Part_Of_View AS
SELECT Race_ID, Championship_ID
FROM Is_Part_Of;
```

```
-- View for User table
CREATE VIEW User_View AS
SELECT User_ID, First_Name, Last_Name, DOB, Phone_No, Email, Username
FROM User;
```

TRIGGERS

1. Trigger to Store Changes in `Team` Table

Whenever a team's name is updated, this trigger will store the old and new names in a separate `Team_Name_Changes` table.

```
CREATE TABLE Team_Name_Changes (
    Change_ID INT AUTO_INCREMENT PRIMARY KEY,
    Old_Team_Name VARCHAR(100),
    New_Team_Name VARCHAR(100),
    Change_Timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

DELIMITER //
CREATE TRIGGER Track_Team_Name_Changes
BEFORE UPDATE ON Team
FOR EACH ROW
BEGIN
    IF OLD.Team_Name <> NEW.Team_Name THEN
        INSERT INTO Team_Name_Changes (Old_Team_Name, New_Team_Name)
        VALUES (OLD.Team_Name, NEW.Team_Name);
    END IF;
END;
//
DELIMITER ;
```

2. Trigger to Store Changes in `Driver` Table (Team Changes)

Whenever a driver switches teams, this trigger will log the change.

```
CREATE TABLE Driver_Team_Changes (
    Change_ID INT AUTO_INCREMENT PRIMARY KEY,
    Driver_ID VARCHAR(7),
    Old_Team_ID VARCHAR(7),
    New_Team_ID VARCHAR(7),
    Change_Timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (Driver_ID) REFERENCES Driver(Driver_ID)
);
```

```
DELIMITER //

CREATE TRIGGER Track_Driver_Team_Changes
BEFORE UPDATE ON Driver
FOR EACH ROW
BEGIN
   IF OLD.Team_ID <> NEW.Team_ID THEN
      INSERT INTO Driver_Team_Changes (Driver_ID, Old_Team_ID, New_Team_ID)
      VALUES (OLD.Driver_ID, OLD.Team_ID, NEW.Team_ID);
   END IF;
END;
//
DELIMITER ;
```

3. Trigger: Track Driver Wins Updates

This trigger logs changes to the number of wins for a driver whenever it gets updated.

```
CREATE TABLE Driver_Wins_Log (
   Log_ID INT AUTO_INCREMENT PRIMARY KEY,
   Driver_ID VARCHAR(7),
   Old_Wins INT,
   New_Wins INT,
   Change_Time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

DELIMITER $$

CREATE TRIGGER Track_Driver_Wins_Update
BEFORE UPDATE ON Driver
FOR EACH ROW
BEGIN
```

```
    IF OLD.Wins <> NEW.Wins THEN
        INSERT INTO Driver_Wins_Log (Driver_ID, Old_Wins, New_Wins)
        VALUES (OLD.Driver_ID, OLD.Wins, NEW.Wins);
    END IF;
END $$

DELIMITER ;
```

CURSORS

1. Cursor to Loop Through Teams and Print Their Wins

This stored procedure will loop through all teams and print their win count.

Cursor Code:

```
DELIMITER //
CREATE PROCEDURE List_Team_Wins()
BEGIN
   DECLARE done INT DEFAULT FALSE;
   DECLARE team_name VARCHAR(100);
   DECLARE wins_count INT;
   DECLARE team_cursor CURSOR FOR SELECT Team_Name, Wins FROM Team;
   DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

   OPEN team_cursor;

   team_loop: LOOP
      FETCH team_cursor INTO team_name, wins_count;
      IF done THEN
         LEAVE team_loop;
      END IF;
      SELECT CONCAT('Team: ', team_name, ' - Wins: ', wins_count) AS Team_Wins_Info;
   END LOOP;

   CLOSE team_cursor;
END;
//
DELIMITER ;
```

To execute the cursor:

```
CALL List_Team_Wins();
```

Expected Output

Team_Wins_Info
Team: Oracle Red Bull Racing - Wins: 11
Team: McLaren F1 Team - Wins: 4
Team: Scuderia Ferrari - Wins: 3
Team: Mercedes-AMG Petronas - Wins: 2

2. Cursor: Get All Races of a Given Driver

This cursor fetches all races that a specific driver participated in.

Cursor Code:

```
DELIMITER //

CREATE PROCEDURE Get_Driver_Races(IN driver_id_param VARCHAR(7))
BEGIN
    DECLARE done INT DEFAULT FALSE;
    DECLARE race_name VARCHAR(100);
    DECLARE driver_cursor CURSOR FOR
        SELECT Race_Name FROM Race WHERE Winner_Driver_ID = driver_id_param;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    OPEN driver_cursor;

    driver_loop: LOOP
        FETCH driver_cursor INTO race_name;
        IF done THEN
            LEAVE driver_loop;
        END IF;
        SELECT CONCAT('Race: ', race_name) AS Driver_Race_Info;
    END LOOP;

    CLOSE driver_cursor;
END;
//
```

DELIMITER ;

To execute the cursor:
To get all races won by Max Verstappen (`D001`):

CALL Get_Driver_Races('D001');

Expected Output

Race_Name
British Grand Prix

QUERIES AND THEIR OUTPUT

1. Queries for `Team` Table

1. Retrieve all teams sorted by the most championship wins:

```
SELECT * FROM Team ORDER BY Championship_Wins DESC;
```
Expected Output:

Team_ID	Team_Name	Wins	Championship_Wins	Year_Founded
T003	Scuderia Ferrari	16	16	1950
T002	McLaren F1 Team	9	9	1966
T001	Oracle Red Bull Racing	5	5	2005
T004	Mercedes-AMG Petronas F1 Team	8	8	2010

Note: Championship wins are cumulative up to and including the 2024 season.

2. Get the total number of wins by all teams:

```
SELECT SUM(Wins) AS Total_Wins FROM Team;
```
Expected Output:

Total_Wins
38

Note: This total reflects the combined wins of the teams up to 2024.

3. Find teams founded before the year 2000:

```
SELECT * FROM Team WHERE Year_Founded < 2000;
```
Expected Output:

Team_ID	Team_Name	Wins	Championship_Wins	Year_Founded
T002	McLaren F1 Team	9	9	1966
T003	Scuderia Ferrari	16	16	1950

4. Find the team with the highest wins:

```
SELECT * FROM Team WHERE Wins = (SELECT MAX(Wins) FROM Team);
```
Expected Output:

Team_ID	Team_Name	Wins	Championship_Wins	Year_Founded
T003	Scuderia Ferrari	16	16	1950

5. Get teams that have never won a championship:

```
SELECT * FROM Team WHERE Championship_Wins = 0;
```
Expected Output:

Team_ID	Team_Name	Wins	Championship_Wins	Year_Founded

Note: All listed teams have won at least one championship by 2024.

2. Queries for `Driver` Table

1. Retrieve all drivers who have won at least one championship:

```
SELECT * FROM Driver WHERE Championship_Wins > 0;
```
Expected Output:

Driver_ID	First_Name	Last_Name	DOB	Wins	Championship_Wins	Did_Not_Finish	Team_ID	Race_Year
D001	Max	Verstappen	1997-09-30	50	4	10	T001	2024
D002	Lando	Norris	1999-11-13	10	1	5	T002	2024
D003	Charles	Leclerc	1997-10-16	5	0	7	T003	2024
D004	Lewis	Hamilton	1985-01-07	103	7	28	T004	2024

Note: As of 2024, Charles Leclerc has not won a championship.

2. Get the average number of wins per driver:

```
SELECT AVG(Wins) AS Avg_Wins FROM Driver;
```
Expected Output:

Avg_Wins
42

3. Find drivers belonging to 'Oracle Red Bull Racing':

```
SELECT * FROM Driver WHERE Team_ID = (SELECT Team_ID FROM Team
WHERE Team_Name = 'Oracle Red Bull Racing');
```
Expected Output:

Driver_ID	First_Name	Last_Name	DOB	Wins	Championship_Wins	Did_Not_Finish	Team_ID	Race_Year
D001	Max	Verstappen	1997-09-30	50	4	10	T001	2024

4. Get the top 3 drivers with the most wins:

```
SELECT * FROM Driver ORDER BY Wins DESC LIMIT 3;
```
Expected Output:

Driver_ID	First_Name	Last_Name	DOB	Wins	Championship_Wins	Did_Not_Finish	Team_ID	Race_Year
D004	Lewis	Hamilton	1985-01-07	103	7	28	T004	2024
D001	Max	Verstappen	1997-09-30	50	4	10	T001	2024
D002	Lando	Norris	1999-11-13	10	1	5	T002	2024

5. Find drivers with more than 5 "Did Not Finish" (DNF) records:

```
SELECT * FROM Driver WHERE Did_Not_Finish > 5;
```
Expected Output:

Driver_ID	First_Name	Last_Name	DOB	Wins	Championship_Wins	Did_Not_Finish	Team_ID	Race_Year
D003	Charles	Leclerc	1997-10-16	5	0	7	T003	2024
D004	Lewis	Hamilton	1985-01-07	103	7	28	T004	2024

3. Queries for `Car_Specs` Table

1. Retrieve all cars associated with 'Scuderia Ferrari':

```
SELECT * FROM Car_Specs WHERE Team_ID = (SELECT Team_ID FROM Team
WHERE Team_Name = 'Scuderia Ferrari');
```
Expected Output:

Car_ID	Chassis	Engine	HP	Torque	Color	Gearbox	Team_ID
C003	SF-24	Ferrari 066/10	1030	550	Red	8-speed	T003

2. Find the car with the highest horsepower:

```
SELECT * FROM Car_Specs WHERE HP = (SELECT MAX(HP) FROM
Car_Specs);
```
Expected Output:

Car_ID	Chassis	Engine	HP	Torque	Color	Gearbox	Team_ID
C001	RB20	Honda RA621H	1040	560	Blue	8-speed	T001

3. Count the number of unique car chassis used in the season:

```
SELECT COUNT(DISTINCT Chassis) FROM Car_Specs;
```
Expected Output:

COUNT(DISTINCT Chassis)
4

4. List all car colors used across teams:

```
SELECT DISTINCT Color FROM Car_Specs;
```
Expected Output:

Color
Blue
Orange
Red
Silver

5. Find the team that uses the most powerful engine:

```
SELECT Team_ID FROM Car_Specs WHERE HP = (SELECT MAX(HP) FROM
Car_Specs);
```
Expected Output:

Team_ID
T001

4. Queries for `Circuit` Table

1. Retrieve all circuits in Europe:

```
SELECT * FROM Circuit WHERE Location LIKE '%Europe%';
```
Expected Output (Example, as "Europe" isn't stored explicitly):

Circuit_ID	Circuit_Name	Location	Length
C001	Silverstone	UK	5.891

2. Find the longest circuit:

```
SELECT * FROM Circuit WHERE Length = (SELECT MAX(Length) FROM
Circuit);
```
Expected Output:

Circuit_ID	Circuit_Name	Location	Length
C002	Interlagos	Brazil	7.004

3. Get the total number of circuits in the database:

```
SELECT COUNT(*) FROM Circuit;
```
Expected Output:

COUNT(*)
2

4. Find circuits that have hosted at least one race:

```
SELECT DISTINCT c.* FROM Circuit c JOIN Race r ON c.Circuit_ID =
r.Circuit_ID;
```
Expected Output:

Circuit_ID	Circuit_Name	Location	Length
C001	Silverstone	UK	5.891
C002	Interlagos	Brazil	7.004

5. Rank circuits by length in descending order:

```
SELECT * FROM Circuit ORDER BY Length DESC;
```
Expected Output:

Circuit_ID	Circuit_Name	Location	Length
C002	Interlagos	Brazil	7.004
C001	Silverstone	UK	5.891

5. Queries for `Race` Table

1. Get all races held in 2024

```
SELECT * FROM Race WHERE Year = 2024;
```
Expected Output:

Race_ID	Race_Name	Circuit_ID	Year	Winner_Driver_ID	Winner_Team_ID
R001	British Grand Prix	C001	2024	D001	T001
R002	São Paulo Grand Prix	C002	2024	D002	T002

2. Find the winner of the British Grand Prix

```
SELECT d.Driver_Name, t.Team_Name
FROM Race r
JOIN Driver d ON r.Winner_Driver_ID = d.Driver_ID
JOIN Team t ON r.Winner_Team_ID = t.Team_ID
```

```
WHERE r.Race_Name = 'British Grand Prix';
```
Expected Output:

Driver_Name	Team_Name
Max Verstappen	Oracle Red Bull Racing

3. Count the total number of races in 2024

```
SELECT COUNT(*) FROM Race WHERE Year = 2024;
```
Expected Output:

COUNT(*)
2

4. Get the circuit names for all races held in 2024

```
SELECT r.Race_Name, c.Circuit_Name
FROM Race r
JOIN Circuit c ON r.Circuit_ID = c.Circuit_ID
WHERE r.Year = 2024;
```
Expected Output:

Race_Name	Circuit_Name
British Grand Prix	Silverstone
São Paulo Grand Prix	Interlagos

5. Get races with winners sorted by team name

```
SELECT r.Race_Name, d.Driver_Name, t.Team_Name
FROM Race r
JOIN Driver d ON r.Winner_Driver_ID = d.Driver_ID
JOIN Team t ON r.Winner_Team_ID = t.Team_ID
ORDER BY t.Team_Name;
```
Expected Output:

Race_Name	Driver_Name	Team_Name
São Paulo Grand Prix	Lando Norris	McLaren F1 Team
British Grand Prix	Max Verstappen	Oracle Red Bull Racing

6. Queries for `Championship` Table

1. Get all championships held in 2024:

```
SELECT * FROM Championship WHERE Year = 2024;
```
Expected Output:

Championship_ID	Year	Type	Winner_Driver_ID	Winner_Team_ID
CH001	2024	Formula 1	D001	T001

2. Find championships won by Ferrari:

```
SELECT * FROM Championship WHERE Winner_Team_ID = (SELECT Team_ID
FROM Team WHERE Team_Name = 'Scuderia Ferrari');
```
Expected Output: (No championships won in 2024)

Championship_ID	Year	Type	Winner_Driver_ID	Winner_Team_ID
NULL	NULL	NULL	NULL	NULL

3. Count the total number of championships held:

```
SELECT COUNT(*) FROM Championship;
```
Expected Output:

COUNT(*)
1

4. List all championship winners sorted by year:

```
SELECT * FROM Championship ORDER BY Year;
```
Expected Output:

Championship_ID	Year	Type	Winner_Driver_ID	Winner_Team_ID
CH001	2024	Formula 1	D001	T001

5. Find the last championship winner:

```
SELECT * FROM Championship ORDER BY Year DESC LIMIT 1;
```
Expected Output: (2024 championship)

Championship_ID	Year	Type	Winner_Driver_ID	Winner_Team_ID
CH001	2024	Formula 1	D001	T001

7. Queries for `Pitstop` Table

1. Get all pit stops from the Silverstone race:

```
SELECT * FROM Pitstop WHERE Race_ID = (SELECT Race_ID FROM Race
WHERE Race_Name = 'British Grand Prix');
```
Expected Output:

P001	R001	D001	1	22.345
P002	R001	D002	2	22.567

2. Find the fastest pit stop in any race:

```
SELECT * FROM Pitstop ORDER BY Lap_Time ASC LIMIT 1;
```
Expected Output:

Pitstop_ID	Race_ID	Driver_ID	Position	Lap_Time
P001	R001	D001	1	22.345

3. Get the average pit stop time per race:

```
SELECT Race_ID, AVG(Lap_Time) AS Avg_Pit_Time FROM Pitstop GROUP
BY Race_ID;
```
Expected Output:

Race_ID	Avg_Pit_Time
R001	22.456

4. Count the total number of pit stops per driver:

```
SELECT Driver_ID, COUNT(*) AS Total_Pitstops FROM Pitstop GROUP BY
Driver_ID;
```
Expected Output:

Driver_ID	Total_Pitstops
D001	1
D002	1

5. Rank drivers by fastest pit stop time:

```
SELECT Driver_ID, MIN(Lap_Time) AS Fastest_Pit_Time FROM Pitstop
GROUP BY Driver_ID ORDER BY Fastest_Pit_Time ASC;
```
Expected Output:

Driver_ID	Fastest_Pit_Time
D001	22.345
D002	22.567

8. Queries for `Qualifying` Table

1. Get the qualifying results for Silverstone:

```
SELECT * FROM Qualifying WHERE Race_ID = (SELECT Race_ID FROM Race
WHERE Race_Name = 'British Grand Prix');
```
Expected Output:

Qualifying_ID	Race_ID	Driver_ID	Position	Lap_Time
Q001	R001	D001	1	85.672
Q002	R001	D002	2	86.123

2. Find the fastest qualifying lap in any race:

```
SELECT * FROM Qualifying ORDER BY Lap_Time ASC LIMIT 1;
```
Expected Output:

Qualifying_ID	Race_ID	Driver_ID	Position	Lap_Time
Q001	R001	D001	1	85.672

9. Queries for `User` Table

1. Get all user details for username 'sarthak':

```
SELECT * FROM User WHERE Username = 'sarthak';
```
Expected Output:

User_ID	First_Name	Last_Name	DOB	Phone_No	Email	Username	Password
U001	Sarthak	Sharma	2004-10-30	9876543210	sarthak@example.com	sarthak	hashed_pw

2. Count total number of registered users:

```
SELECT COUNT(*) FROM User;
```
Expected Output:

COUNT(*)
1

3. Find users above 18 years old:

```
SELECT * FROM User WHERE YEAR(CURDATE()) - YEAR(DOB) >= 18;
```
Expected Output:

User_ID	First_Name	Last_Name	DOB	Phone_No	Email	Username	Password

U001	Sarthak	Sharma	2004-10-30	9876543210	sarthak@example.com	sarthak	hashed_pw

4. Get users with unique email domains:

```
SELECT DISTINCT SUBSTRING_INDEX(Email, '@', -1) AS Domain FROM User;
```

Expected Output:

Domain
example.com

5. Retrieve users sorted by age (oldest first):

```
SELECT * FROM User ORDER BY DOB ASC;
```

Expected Output:

User_ID	First_Name	Last_Name	DOB	Phone_No	Email	Username	Password
U001	Sarthak	Sharma	2004-10-30	9876543210	sarthak@example.com	sarthak	hashed_pw