Question 1 - team members
Christine Liu and Stephen Zhang
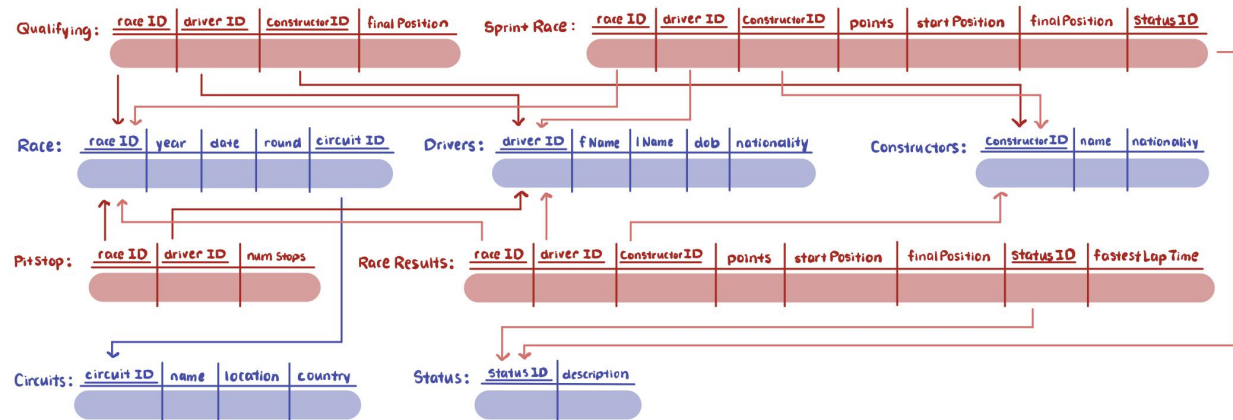

Question 2 - target domain
Formula One Database

Question 3 - sample questions
1. Return nationalities and driver count. Order by descending number of drivers.
2. Return the first name and last name of all drivers of a specific nationality.
3. Return countries and circuit count. Order by descending number of circuits.
4. Return the name and location of all circuits in a specific country.
5. Return the first name and last name of all drivers who have won a race.
6. Return the first name and last name of all drivers who have gotten pole from qualifying.
7. Return the first name and last name of all drivers who have won a sprint race.
8. Return the average number points for a specific circuit of each driver and the respective driver's first name and last name. Order by descending number of points.
9. Return the average number of points per season of each driver and the respective driver's first name and last name. Order by descending number of points.
10. Return the average number points for a specific circuit of each constructor and the respective constructor's name. Order by descending number of points.
11. Return the average number of points per season of each constructor and the respective constructor's name. Order by descending number of points.
12. Return driver first name and last names, and their respective number of wins in a specific circuit. Order by descending number of wins.
13. Return driver first name and last names, and their respective fastest lap time in a specific circuit. Order by ascending all time fastest lap time.
14. Return the name and location of all circuits and it's all time total race accidents/collisions. Order by descending accidents.
15. Return the name and location of all circuits and its average accidents/collisions per race. Order by descending average accidents.
16. Return first name and last name of all drivers who have ever had an accident/collision in a specific circuit.
17. Return the first name and last name of all drivers, and their all time number of incomplete races (due to accident/collision, mechanical failures, etc). Order by descending number of incomplete races.
18. Return average number of pit stops of all drivers that have won at a specific circuit. Order by ascending pit stops.
19. Return nationalities and all its drivers' average number of points per season. Order by descending points.
20. Return date of birth ranges and all its drivers' average number of points per season. Order by descending points.

Question 4 - data model & implementation

Qualifying: | race ID | driver ID | ConstructorID | final Position

Sprint Race: | race ID | driver ID | ConstructorID | points | start Position | final Position | Status ID

Race: | race ID | year | date | round | circuit ID

Drivers: | driver ID | f Name | l Name | dob | nationality

Constructors: | ConstructorID | name | nationality

PitStop: | race ID | driver ID | num Stops

Race Results: | race ID | driver ID | ConstructorID | points | start Position | final Position | Status ID | fastest Lap Time

Circuits: | circuit ID | name | location | country

Status: | Status ID | description

See SQL file for implementation.

Question 5 - sample SQL statements
1) SELECT nationality, COUNT(driverID)
FROM Drivers
GROUP BY nationality
ORDER BY COUNT(nationality) DESC;

5) SELECT DISTINCT D.fname, D.lname
FROM Drivers AS D, RaceResults AS R
WHERE D.driverID = R.driverID AND
      D.position = 1;

8) SELECT D.fname, D.lName, AVG(R.points)
FROM Driver AS D, RaceResults AS R, Circuit AS C
WHERE C.name = "Hockenheimring" AND
      D.driverID = R.driverID AND
      C.circuitID = R.circuitID AND
GROUP BY D .driverID
ORDER BY AVG(R.points)

12) WITH HockCircuit AS(SELECT C.circuitID
      FROM Circuits AS C
      WHERE C.name = "Hockenheimring"),
DriversInHockCircuit AS (SELECT D.driverID, D.fName, D.lName
      FROM Qualifying AS Q JOIN Race AS R ON Q.raceID = R.raceID JOIN HockCircuit AS
HC ON R.circuitID = HC.circuitID JOIN Drivers AS D ON Q.driverID = D.driverID),
SELECT DHC.fName, DHC.lName, COUNT(RR.finalPosition) AS totalWins
FROM DriversInHockCircuit AS DHC JOIN RaceResults AS RR ON DHC.driverID =
RR.driverID

WHERE RR.finalPosition = 1
GROUP BY RR.driverID
ORDER BY COUNT(RR.finalPosition) DESC;

14) SELECT C.name, C.location, COUNT(C.circuitID)
FROM Circuit AS C, RaceResults AS R
WHERE C.circuitID = R.circuitID AND
        (R.status = 3 OR R.status = 4)
GROUP BY C.circuitID
ORDER BY COUNT(C.circuitID) DESC;

Question 6 - plan to load database
Data is extracted from:
https://www.kaggle.com/datasets/thedevastator/formula-one-racing-a-comprehensive-data-analysis
We will have to modify .csv files to follow our database implementation.

Question 7 - views & output
We plan to display results that will bring interesting statistics to the user. For example, obtaining the average number of points of all drivers from a specific nationality or the average number of points that a driver has on each circuit. We plan to generate more results along the lines of showing the averages for different attributes and the total amount of an attribute. There will be a table, and perhaps a graph to help visualize our data if it is ordered.

One view we will create is that for every pair of races and drivers there will be a row that consists of all driver information, the circuit, constructor, and points. This will be helpful in calculating point averages. We may create more views as it seems fit.

Question 8 - specialized/advanced topics
Particularly advanced GUI form interface and/or report generation