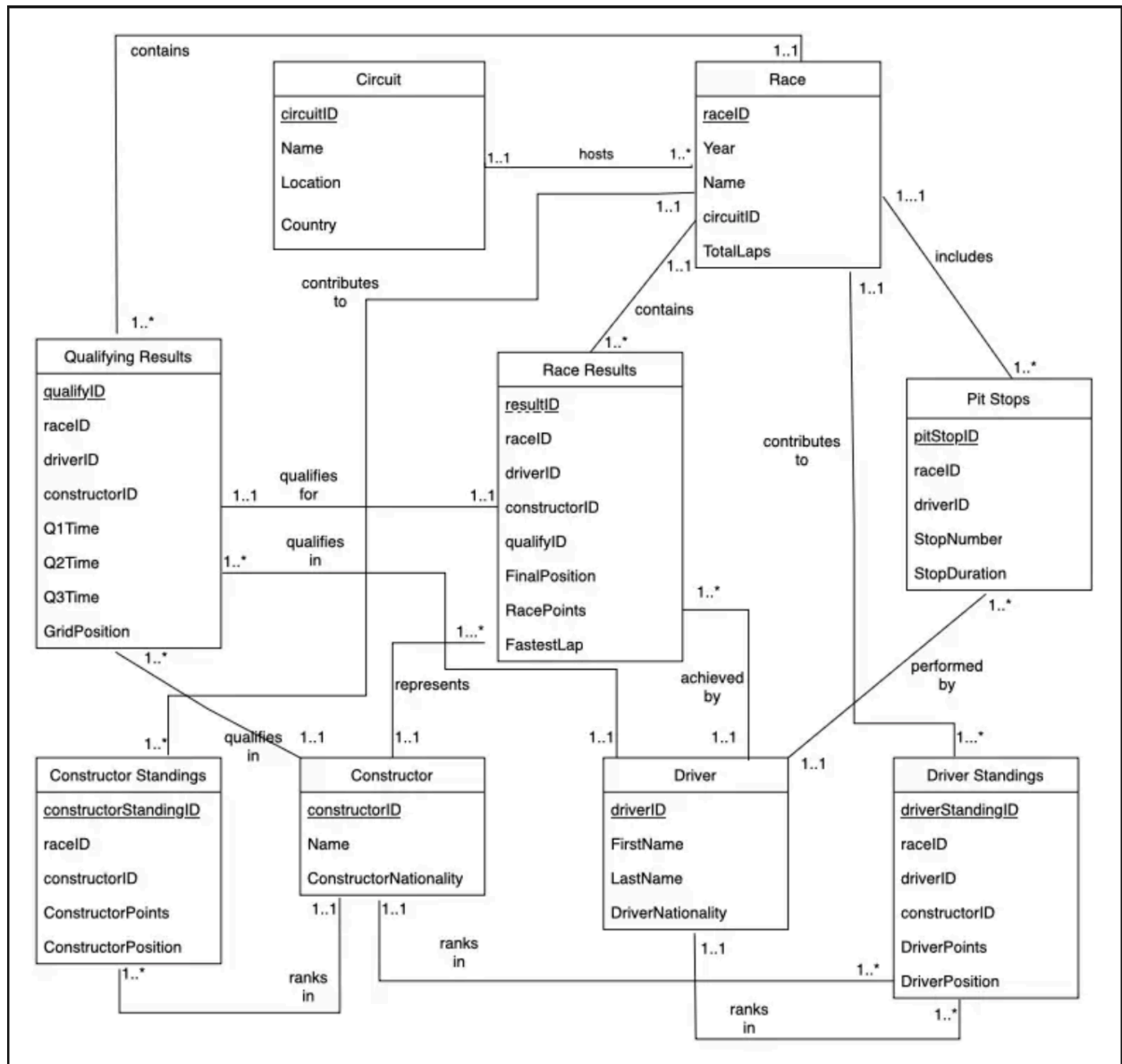


## UML Diagram:



### **Assumptions:**

**Circuit:** For each circuit, at least one race has been hosted on that circuit. Each circuit has a unique identifier and contains the name, location, and country attributes.

**Race:** Each race is associated with exactly one circuit. A race can have multiple results, qualifying results, and pit stops. Every race belongs to a specific year and has a unique race ID.

**Qualifying Results:** For each race, there is one set of qualifying results. A driver and constructor participate in these qualifications

**Race Results:** Each driver who participates in a race has exactly one race result, which includes the final position, race points, and whether they set the fastest lap. Every result is tied to one qualifying result and involves one constructor.

**Pit Stops:** Each driver may have multiple pit stops in a race. Every pit stop is tied to one race, with details like stop number and stop duration recorded.

**Constructor:** Each constructor participates in multiple races, and each constructor has multiple results and standings in different races.

**Constructor Standings:** For every race, there is a record of constructor standings, where each constructor earns points and is ranked based on total points scored.

**Driver:** Each driver can participate in multiple races, and for each race, they can have one set of results. Drivers are associated with specific constructors for each race but can be associated with different constructors across races.

**Driver Standings:** For each race, there is a record of driver standings. Drivers earn points based on their performance and are ranked accordingly for each race.

## Cardinality

- **Circuit to Race:** one-to-many (One circuit hosts many races)
- **Race to Race Results:** one-to-many (One race has many race results)
- **Race to Pit Stops:** one-to-many (One race includes many pit stops)
- **Race to Qualifying Results:** one-to-many (One race has many qualifying results)
- **Qualifying Results to Driver:** many-to-one (Many qualifying results are associated with one driver)
- **Qualifying Results to Constructor:** many-to-one (Many qualifying results are associated with one constructor)
- **Race Results to Driver:** many-to-one (Many race results are associated with one driver)
- **Race Results to Constructor:** many-to-one (Many race results are associated with one constructor)
- **Race Results to Qualifying Results:** one-to-one (One race result corresponds to one qualifying result)
- **Pit Stops to Driver:** many-to-one (Many pit stops are performed by one driver)
- **Constructor to Constructor Standings:** one-to-many (One constructor has many constructor standings)
- **Driver to Driver Standings:** one-to-many (One driver has many driver standings)
- **Race to Driver Standings:** one-to-many (One race includes many driver standings)
- **Race to Constructor Standings:** one-to-many (One race includes many constructor standings)

## Description:

1. **hosts:** Each Circuit hosts at least one Race, while each Race is hosted by exactly one Circuit.
2. **contains:** Each Race contains multiple Race Results, while each Race Result is associated with exactly one Race.
3. **includes:** Each Race includes multiple Pit Stops, while each Pit Stop is associated with exactly one Race.
4. **performed by:** Each Pit Stop is performed by exactly one Driver, while each Driver may perform multiple Pit Stops across different races.
5. **contributes to (Race Results to Race):** Each Race Result contributes to exactly one Race, while each Race may have multiple Race Results.
6. **achieved by:** Each Race Result is achieved by exactly one Driver, while each Driver may achieve multiple Race Results across different races.
7. **represents:** Each Race Result represents exactly one Constructor, while each Constructor may have multiple Race Results across different races.

8. **qualifies for**: Each set of Qualifying Results qualifies for exactly one Race, while each Race may have multiple sets of Qualifying Results.
9. **qualifies in (Driver)**: Each Driver qualifies in exactly one set of Qualifying Results, while each set of Qualifying Results involves multiple Drivers.
10. **qualifies in (Constructor)**: Each Constructor qualifies in exactly one set of Qualifying Results, while each set of Qualifying Results involves multiple Constructors.
11. **ranks in (Driver Standings)**: Each Driver ranks in multiple Driver Standings, while each Driver Standing is associated with exactly one Driver.
12. **ranks in (Constructor Standings)**: Each Constructor ranks in multiple Constructor Standings, while each Constructor Standing is associated with exactly one Constructor.
13. **contributes to (Race Results to Standings)**: Each Race Result contributes to multiple Driver Standings and Constructor Standings, while each Driver Standing and Constructor Standing depends on multiple Race Results.

### **Functional Dependencies**

- **Circuit**: circuitID  $\rightarrow$  Name, Location, Country
- **Race**: raceID  $\rightarrow$  Year, Name, circuitID, TotalLaps
- **Qualifying Results**: qualifyID  $\rightarrow$  raceID, driverID, constructorID, Q1Time, Q2Time, Q3Time, GridPosition
- **Race Results**: resultID  $\rightarrow$  raceID, driverID, constructorID, qualifyID, FinalPosition, RacePoints, FastestLap
- **Pit Stops**: pitStopID  $\rightarrow$  raceID, driverID, StopNumber, StopDuration
- **Constructor Standings**: constructorStandingID  $\rightarrow$  raceID, constructorID, ConstructorPoints, ConstructorPosition
- **Driver Standings**: driverStandingID  $\rightarrow$  raceID, driverID, constructorID, DriverPoints, DriverPosition
- **Constructor**: constructorID  $\rightarrow$  Name, ConstructorNationality
- **Driver**: driverID  $\rightarrow$  FirstName, LastName, DriverNationality

## Normalization

- **Circuit**( circuitID, Name, Location, Country )
- **Race**( raceID, Year, Name, circuitID, TotalLaps )
- **Qualifying Results**( qualifyID, raceID, driverID, constructorID, Q1Time, Q2Time, Q3Time, GridPosition )
- **Race Results**( resultID, raceID, driverID, constructorID, qualifyID, FinalPosition, RacePoints, FastestLap )
- **Pit Stops**( pitStopID, raceID, driverID, StopNumber, StopDuration )
- **Constructor Standings**( constructorStandingID, raceID, constructorID, ConstructorPoints, ConstructorPosition )
- **Driver Standings**( driverStandingID, raceID, driverID, constructorID, DriverPoints, DriverPosition )
- **Constructor**( constructorID, Name, ConstructorNationality )
- **Driver**( driverID, FirstName, LastName, DriverNationality )

## Why 3NF?

1. **Eliminates Redundancy**: Reduces duplicate data by ensuring no transitive dependencies.
2. **Improves Data Integrity**: Prevents update anomalies and maintains consistency.
3. **Efficient Storage**: Minimizes data duplication, saving space

## We choose 3NF over BCNF because:

1. **Simplicity & Fewer Constraints**:: 3NF is easier to understand and implement. BCNF can be more restrictive and harder to maintain when complex relationships exist. Moreover 3NF allows for more flexibility without sacrificing data integrity.

## Relational Schema

**Circuit**: (  
circuitID: INT [PK],  
Name: VARCHAR(100),  
Location: VARCHAR(100),  
Country: VARCHAR(50)  
)

**Race:** (  
raceID: INT [PK],  
Year: INT,  
Name: VARCHAR(100),  
circuitID: INT [FK to Circuit.circuitID],  
TotalLaps: INT  
)

**Qualifying\_Results:** (  
qualifyID: INT [PK],  
raceID: INT [FK to Race.raceID],  
driverID: INT [FK to Driver.driverID],  
constructorID: INT [FK to Constructor.constructorID],  
Q1Time: TIME,  
Q2Time: TIME,  
Q3Time: TIME,  
GridPosition: INT  
)

**Race\_Results:** (  
resultID: INT [PK],  
raceID: INT [FK to Race.raceID],  
driverID: INT [FK to Driver.driverID],  
constructorID: INT [FK to Constructor.constructorID],  
qualifyID: INT [FK to Qualifying\_Results.qualifyID],  
FinalPosition: INT,  
RacePoints: INT,  
FastestLap: TIME  
)

**Pit\_Stops:** (  
pitStopID: INT [PK],  
raceID: INT [FK to Race.raceID],  
driverID: INT [FK to Driver.driverID],  
StopNumber: INT,  
StopDuration: TIME  
)

**Constructor\_Standings:** (  
constructorStandingID: INT [PK],  
raceID: INT [FK to Race.raceID],  
constructorID: INT [FK to Constructor.constructorID],  
ConstructorPoints: INT,  
ConstructorPosition: INT  
)

**Driver\_Standings:** (  
driverStandingID:INT [PK],  
raceID:INT [FK to Race.raceID],  
driverID:INT [FK to Driver.driverID],  
constructorID:INT [FK to Constructor.constructorID],  
DriverPoints:INT,  
DriverPosition:INT  
)

**Constructor:** (  
constructorID: INT [PK],  
Name: VARCHAR(100),  
ConstructorNationality: VARCHAR(50)  
)

**Driver:** (  
driverID:INT [PK],  
FirstName:VARCHAR(50),  
LastName:VARCHAR(50),  
  
DriverNationality:VARCHAR(50)  
)