Milestone 1

University of British Columbia, Vancouver
Department of Computer Science
CPSC 304 Project - The Formula for Success
Milestone #: 1
Date: October 6, 2023
Group Number: 51

Name	Student Number	CS Alias	Email Address
Kira Swinth	38122560	i9b6u	kira.swinth@gmail.com
Kelly Zhang	46990602	b8g1d	kellyz02@student.ubc.ca
Faraneh Yahyaei-Moayyed	60431905	x3y9g	faranehyahyaei@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your email address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Project Description:

What is the domain of the application?

The domain of our application is sports data management. More specifically, our application focuses on Formula One (F1), a form of international motorsport (see *Further Comments* for more information about F1). The application will store statistics and results for fans to reference and explore.

What aspects of the domain are modeled by the database?

The database will model important components of the F1 system including Drivers, Constructors (i.e. "teams"), Circuits, Races, Standings and additional ones to accurately represent the intricacies of F1's structure. In addition, the database will model relationships between them such as which races drivers compete in, the results of those races, the overall performance of a team in a given season, and many more. Our project focuses on these core components of F1 as the database is intended to support fan engagement and activities, such as F1 betting and fantasy leagues.

Database Specifications:

What functionality will the database provide?

The database will store information about the central components of F1 such as drivers and constructors. This will allow for the exploration of statistics between drivers, across teams, across circuits, across seasons, etc. Some examples of driver statistics include total points, points per race, and current position in the standings. The database will allow for fans to analyze trends such as how positions and performance of the cars are changing. We are also including details about technical aspects of the cars so fans can see how design can affect performance. Unlike other F1 analytics platforms, users can also view information about team sponsors and viewership statistics per Grand Prix (race).

Application Platform:

We will be using the department provided Oracle for the DBMS, PHP for programming the backend of our database, and CSS or HTML for the frontend implementation. We will also be using Github to support our collaboration and progress.

Further Comments:

What is Formula One (F1)? F1 is arguably the highest class of motorsport (car racing) in the world. There are 10 teams (called "constructors") in the competition each year. Each team has two drivers who drive the race cars and hundreds of employees who engineer the car, build the car, strategize the races, and market the team. The cars race at 23 different race tracks ("circuits") around the world each year. Each race is called a Grand Prix. The car that finishes the Grand Prix in the quickest time wins (often referred to as "P1"). A driver's finishing place determines how many points they get. Constructors and drivers accumulate points over the course of all the Grand Prixs. At the end of the year, the constructor with the most points wins the F1 Constructor's Championship, and the driver with the most points wins the F1 Driver's Championship.

The sport was initially popularized in Europe but has been expanding its audience in recent years to Asia and North America. The increasing audience has also led to increasing money in the sport. Companies can sponsor teams to have their brand be visible on the car. The increasing audience has also led to an increase in media attention to the personal lives of the drivers. One area of frequent attention is the dating lives of the drivers. The partners have become a central part of the drivers’ celebrity status. For more information about the structure of F1, check out this fun video.

1. *Makes - combined with the “many” side (Car relation), as the relationship is Car → Constructor*
2. *Uses - combined with the “many” side (GrandPrix relation), as the relationship is GrandPrix → Circuit*
3. *HasConstructor modelled by weak entity relation GrandPrix_ConstrutorStanding*

HasDriver modelled by weak entity relation GrandPrix_DriverStanding

Milestone 2

University of British Columbia, Vancouver
Department of Computer Science
CPSC 304 Project - The Formula for Success
Milestone #: 2
Date: October 19, 2023
Group Number: 51

Name	Student Number	CS Alias	Email Address
Kira Swinth	38122560	i9b6u	kira.swinth@gmail.com
Kelly Zhang	46990602	b8g1d	kellyz02@student.ubc.ca
Faraneh Yahyaei-Moayyed	60431905	x3y9g	faranehyahyaei@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your email address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Summary:

The domain of our application is sports data management. More specifically, our application focuses on Formula One (F1), a form of international motorsport. The application will store statistics and results for fans to reference and explore.

ER Diagram:

Notes on minimal changes made to the ER Diagram since Milestone 1:

- We realized we wanted to make the SponsorshipAmount specific to the "Sponsors" relation between Sponsor and Constructor, as Sponsors can donate different amounts to different Constructors. This involved:
  - Moving "SponsorshipAmount" attribute from "Sponsor" entity to "Sponsors" relation
  - Adding "Industry" attribute to the "Sponsor" entity, so the entity would have more than 1 attribute
- Added attributes to "InRelationshipWith" relationship in order to create more meaningful functional dependencies (other than our primary keys).

Relational Schema

Notes:

- *PKs are underlined; FKs are bolded; CKs and other constraints are indicated*
- *Note: the "YEAR" SQL date type was used for certain attributes, as listed here: https://www.w3schools.com/sql/sql_datatypes.asp*
- *Attributes have been formatted to be lower camel case in our relations*

Entities:

1. Sponsor(companyName: varchar, industry: varchar)
2. Constructor(constructorName: varchar, nationality: varchar, #OfWins: integer)
   - NOT NULL: nationality, #OfWins
   - DEFAULT 0: #OfWins
3. TeamMember(employeeId: integer, firstName: varchar, lastName: varchar, nationality: varchar, dateOfBirth: date, salary: integer, job: varchar)
   - NOT NULL: firstName, lastName, nationality, dateOfBirth, job
4. Car(model: varchar, engine: varchar, **constructorName: varchar**)
   - NOT NULL: constructorName
5. Partner(partnerId: integer, partnerName: varchar, instagramHandle: varchar, instagramFollowers: integer)
   - NOT NULL: partnerName
   - UNIQUE: instagramHandle
   - CK: instagramHandle
     - We decided against using instagramHandle as the PK as it is possible for a Partner to not have an Instagram account.
6. GrandPrix(year: year, gpName: varchar, city: varchar, country: varchar, viewership: integer, attendance: integer, **circuitName: varchar**)
   - NOT NULL: city, country, circuitName

7. Circuit(circuitName: varchar, #OfLaps: integer, length: integer, type: varchar)

Weak Entities:

8. GrandPrix_ConstructorStanding(position: integer, **gpName: varchar**, **year: year**, points: integer)
    - NOT NULL: points
    - DEFAULT 0: points
9. GrandPrix_DriverStanding(racePosition: integer, **gpName: varchar**, **year: year**, points: integer, qualifyingPosition: integer)
    - NOT NULL: points
    - DEFAULT 0: points

ISA:

10. Driver(**employeeId: integer**, #OfPodiums: integer, #OfWins: integer, driverNumber: integer, #OfPolePositions: integer)
    - *Note: Driver ISA has been modelled using "Method 2" as described in lecture, where we have a table for both the superclass (TeamMember) and the subclass (Driver). The primary key of TeamMember, employeeId, serves as the primary key and foreign key for Driver. We chose this method as it allows us to better represent the Drives, InRelationshipWith, and DriverHolds relationships that exclusively involves the Driver entity.*
    - DEFAULT 0: #OfPodiums, #OfWins, #OfPolePositions
    - NOT NULL: #OfPodiums, #OfWins, #OfPolePositions, driverNumber

Relationships:

11. Sponsors(**companyName: varchar**, **constructorName: varchar,** sponsorshipAmount: integer)
12. WorksWith(**constructorName: varchar**, **employeeId: integer**, since: date)
    - *Note: even though constructorName and employeeId are primary keys and therefore, cannot be null, this is not enough to enforce total participation on both sides at this point in time. We acknowledge that this will require using an SQL assertion in the future.*
13. Drives(**model: varchar**, **employeeId: integer**)
    - *Note: even though model and employeeId are primary keys and therefore, cannot be null, this is not enough to enforce total participation on both sides at this point in time. We acknowledge that this will require using an SQL assertion in the future.*
14. InRelationshipWith(**partnerId: integer, employeeId: integer,** since: date**)**
15. ConstructorHolds(**position: integer**, **gpName: varchar**, **year: year, constructorName: varchar**)
16. DriverHolds(**racePosition: integer**, **gpName: varchar**, **year: year**, **employeeId: integer**)

Functional Dependencies:

1. Sponsor(companyName, industry)
   - companyName → industry
2. Constructor(constructorName, nationality, #OfWins)
   - constructorName → nationality, #OfWins
3. TeamMember(employeeId, firstName, lastName, nationality, dateOfBirth, salary, job)
   - employeeId → firstName, lastName, nationality, dateOfBirth, salary, job
4. Car(model, engine, **constructorName**)
   - model → engine, constructorName
5. Partner(partnerId, partnerName, instagramHandle, instagramFollowers)
   - partnerId → partnerName, instagramHandle, instagramFollowers
   - instagramHandle → instagramFollowers
6. GrandPrix(year, gpName, city, country, viewership, attendance, **circuitName**)
   - year, gpName → city, country, viewership, attendance, circuitName
   - circuitName → city, country
   - year, circuitName → attendance, viewership
7. Circuit(circuitName, #OfLaps, length, type)
   - circuitName → #OfLaps, length, type
   - #OfLaps → Length
8. GrandPrix_ConstructorStanding(position, **gpName, year**, points)
   - position, gpName, year → points
   - position → points
9. GrandPrix_DriverStanding(racePosition, **gpName, year**, points, qualifyingPosition)
   - racePosition, gpName, year → points, qualifyingPosition
   - racePosition → points
10. Driver(**employeeId**, #OfPodiums, #OfWins, driverNumber, #OfPolePosition)
    - employeeId → #OfPodiums, #OfWins, driverNumber, #OfPolePositions
11. Sponsors(**companyName, constructorName**, sponsorshipAmount)
    - companyName, constructorName → sponsorshipAmount
12. WorksWith(**constructorName,** **employeeId,** since)
    - constructorName, employeeId → since
13. Drives(**model, employeeId**)
    - no non-trivial FDs
14. InRelationshipWith(**partnerID, employeeID**, since)
    - partnerID, employeeId → since
15. ConstructorHolds(**position, gpName, year, constructorName**)
    - no non-trivial FDs
16. DriverHolds(**racePosition, gpName, year, employeeID**)

- no non-trivial FDs

Normalization:

The process of decomposing to 3NF. PKs are underlined, FK are bolded. For our final, resultant relations with domains and constraints, please see the relations listed after all the decompositions.

Decomposition of Partner Relation

1. Decompose Partner(partnerId, partnerName, instagramHandle, instagramFollowers) on instagramHandle → instagramFollowers:
   - Partner_Ref(partnerId, partnerName, instagramHandle)
   - Partner_2(instagramHandle, instagramFollowers)

Resultant relations: Partner_Ref(partnerId, partnerName, instagramHandle), Partner_2(**instagramHandle**, instagramFollowers)

Decomposition of GrandPrix Relation

1. Create minimal cover for non-BCNF FDs:
   - circuitName → city, country
     - circuitName → city
     - circuitName → country
   - year, circuitName → attendance, viewership
     - year, circuitName → attendance
     - year, circuitName → viewership
2. Decompose GrandPrix(year, gpName, city, country, viewership, attendance, **circuitName**) on circuitName → city
   - GrandPrix_1(**circuitName**, city)
   - GrandPrix_2(year, gpName, country, viewership, attendance, **circuitName**)
3. Decompose GrandPrix_2(year, gpName, country, viewership, attendance, **circuitName**) on circuitName → country
   - GrandPrix_3(**circuitName**, country)
   - GrandPrix_4(year, gpName, viewership, attendance, **circuitName**)
4. Decompose GrandPrix_4(year, gpName, viewership, attendance, **circuitName**) on year, circuitName → attendance
   - GrandPrix_5(year, **circuitName**, attendance)
   - GrandPrix_6(year, gpName, **circuitName**, viewership)
5. Decompose GrandPrix_6(year, gpName, **circuitName**, viewership) on year, circuitName → viewership
   - GrandPrix_Ref(year, **circuitName**, viewership)
   - GrandPrix_8(year, gpName, **circuitName**)

Resultant relations: GrandPrix_1(**circuitName**, city), GrandPrix_3(**circuitName**, country), GrandPrix_5(**year**, **circuitName**, attendance), GrandPrix_Ref(year, **circuitName**, viewership), GrandPrix_8(**year**, gpName, **circuitName**)

For ease of understanding in downstream applications, we have renamed the resultant relations:
~~GrandPrix_2(**circuitName**, city), GrandPrix_3(**circuitName**, country), GrandPrix_4(**year, circuitName**, attendance), GrandPrix_Ref(year, **circuitName**, viewership), GrandPrix_5(**year**, gpName, **circuitName**)~~

Switching GrandPrix_2 + GrandPrix_Ref (as of Milestone 4):
GrandPrix_2(year, **circuitName**, viewership), GrandPrix_3(**circuitName**, country), GrandPrix_4(**year, circuitName**, attendance), GrandPrix_Ref(**circuitName**, city), GrandPrix_5(**year**, gpName, **circuitName**)

Decomposition of Circuit Relation

1. Decompose Circuit(circuitName, #OfLaps, length, type) on #OfLaps → length
   - Circuit_1(circuitName, #OfLaps, type)
   - Circuit_Ref (#OfLaps, length)

Resultant relations: Circuit_Ref (#OfLaps, length), Circuit_2(circuitName, **#OfLaps**, type),

Decomposition of GrandPrix_ConstructorStanding

1. Decompose GrandPrix_ConstructorStanding(position, **gpName, year**, points) on position → points:
   - GrandPrix_ConstructorStanding_2(**position, gpName, year**)
   - GrandPrix_ConstructorStanding_Ref(**position**, points)

Resultant relations: GrandPrix_ConstructorStanding_2(**position, gpName, year**), GrandPrix_ConstructorStanding_Ref(position, points)

Decomposition of GrandPrix_DriverStanding

1. Decompose GrandPrix_DriverStanding(racePosition, **gpName, year**, points, qualifyingPosition) on racePosition → points:
   - GrandPrix_DriverStanding_2(**racePosition, gpName, year,** qualifyingPosition)
   - GrandPrix_DriverStanding_Ref(**racePosition**, points)

Resultant relations: GrandPrix_DriverStanding_2(**racePosition, gpName, year,** qualifyingPosition), GrandPrix_DriverStanding_Ref(racePosition, points)

Normalized Resulting Relations

PKs are underlined; FKs are bolded; CKs and other constraints are indicated

1. Sponsor(companyName: varchar, industry: varchar)
2. Constructor(constructorName: varchar, nationality: varchar, #OfWins: integer)
   - NOT NULL: nationality, #OfWins
   - DEFAULT 0: #OfWins
3. TeamMember(employeeId: integer, firstName: varchar, lastName: varchar, nationality: varchar, dateOfBirth: date, salary: integer, job: varchar)
   - NOT NULL: firstName, lastName, nationality, dateOfBirth, job
4. Car(model: varchar, engine: varchar, **constructorName: varchar**)
   - NOT NULL: constructorName
5. Partner_Ref(partnerId: integer, partnerName: varchar, instagramHandle: varchar)
   - NOT NULL: partnerName

    - UNIQUE: instagramHandle
    - CK: instagramHandle
        i. We decided against using instagramHandle as the PK as it is possible for a Partner to not have an Instagram account.

6. Partner_2(**instagramHandle: varchar**, instagramFollowers: integer)
7. GrandPrix_2(year: year, **circuitName: varchar**, viewership: integer)
8. GrandPrix_Ref(**circuitName: varchar**, city: varchar)
    - NOT NULL: city
9. GrandPrix_3(**circuitName: varchar**, country: varchar)
    - NOT NULL: country
10. GrandPrix_4(**year: year**, **circuitName: varchar**, attendance: integer)
11. GrandPrix_5(**year: year**, gpName: varchar, **circuitName: varchar**)
    - NOT NULL: circuitName
12. Circuit_Ref(#OfLaps: integer, length: integer)
13. Circuit_2(circuitName: varchar, **#OfLaps**: **integer,** type: varchar)
14. GrandPrix_ConstructorStanding_Ref(position: integer, points: integer)
    - NOT NULL: points
    - DEFAULT 0: points
15. GrandPrix_ConstructorStanding_2(**position: integer**, **gpName: varchar**, **year: year**)
16. GrandPrix_DriverStanding_Ref(racePosition: integer, points: integer)
    - NOT NULL: points
    - DEFAULT 0: points
17. GrandPrix_DriverStanding_2(**racePosition: integer**, **gpName: varchar**, **year: year**, qualifyingPosition: integer)
18. Driver(**employeeId: integer**, #OfPodiums: integer, #OfWins: integer, driverNumber: integer, #OfPolePositions: integer)
    - *Note: Driver ISA has been modelled using "Method 2" as described in lecture, where we have a table for both the superclass (TeamMember) and the subclass (Driver). The primary key of TeamMember, employeeId, serves as the primary key and foreign key for Driver. We chose this method as it allows us to better represent the Drives, InRelationshipWith, and DriverHolds relationships that exclusively involves the Driver entity.*
    - DEFAULT 0: #OfPodiums, #OfWins, #OfPolePositions
    - NOT NULL: #OfPodiums, #OfWins, #OfPolePositions, driverNumber
19. Sponsors(**companyName: varchar**, **constructorName: varchar,** sponsorshipAmount: integer)
20. WorksWith(**constructorName: varchar**, **employeeId: integer**, since: date)
    - *Note: even though constructorName and employeeId are primary keys and therefore, cannot be null, this is not enough to enforce total participation on both sides at this point in time. We acknowledge that this will require using an SQL assertion in the future.*
21. Drives(**model: varchar**, **employeeId: integer**)
    - *Note: even though model and employeeId are primary keys and therefore, cannot be null, this is not enough to enforce total participation on both sides at this point in time. We acknowledge that this will require using an SQL assertion in the future.*
22. InRelationshipWith(**partnerId**: **integer**, **employeeId: integer**, since: date**)**

23. ConstructorHolds(**position: integer**, **gpName: varchar**, **year: year**, **constructorName: varchar**)
24. DriverHolds(**racePosition: integer**, **gpName: varchar**, **year: year**, **employeeId: integer**)

SQL DDL:

***Note**: for some attributes, both DEFAULT 0 and NOT NULL constraints were added, as the default value does not prevent user from inserting a NULL value and would only enforce non-null values during the creation of a new tuple based on the discussion here:*
https://stackoverflow.com/questions/11862188/sql-column-definition-default-value-and-not-null-redundant

1. Sponsor(companyName: varchar, industry: varchar)

```
CREATE TABLE Sponsor (
        companyName             varchar             PRIMARY KEY,
        industry                varchar
);
```

2. Constructor(constructorName: varchar, nationality: varchar, #OfWins: integer)

```
CREATE TABLE Constructor (
        constructorName         varchar             PRIMARY KEY,
        nationality             varchar             NOT NULL,
        #OfWins                 int                 DEFAULT 0 NOT NULL
);
```

3. TeamMember(employeeId: integer, firstName: varchar, lastName: varchar, nationality: varchar, dateOfBirth: date, salary: integer, job: varchar)

```
CREATE TABLE TeamMember (
        employeeId              int                 PRIMARY KEY,
        firstName               varchar             NOT NULL,
        lastName                varchar             NOT NULL,
        nationality             varchar             NOT NULL,
        dateOfBirth             date                NOT NULL,
        salary                  int,
        job                     varchar             NOT NULL
);
```

4. Car(model: varchar, engine: varchar, **constructorName: varchar**)

```
CREATE TABLE Car (
        Model                   varchar             PRIMARY KEY,
        engine                  varchar,
        constructorName         varchar             NOT NULL,
      FOREIGN KEY (constructorName) REFERENCES Constructor(constructorName)
            ON DELETE NO ACTION
);
```

5. Partner_Ref(partnerID: integer, partnerName: varchar, instagramHandle: varchar)

```
CREATE TABLE Partner_Ref (
      partnerId               int               PRIMARY KEY,
      partnerName             varchar           NOT NULL,
      instagramHandle         varchar           UNIQUE
);
```

6. Partner_2(**instagramHandle: varchar**, instagramFollowers: integer)

```
CREATE TABLE Partner_2 (
      instagramHandle         varchar           PRIMARY KEY,
      instagramFollowers      int,
      FOREIGN KEY (instagramHandle) REFERENCES Partner_Ref(instagramHandle)
            ON DELETE CASCADE

);
```

7. GrandPrix_2(year: year, **circuitName: varchar**, viewership: integer)

```
CREATE TABLE GrandPrix_Ref (
      year                    year,
      circuitName             varchar,
      viewership              int,
      PRIMARY KEY (year, circuitName),
      FOREIGN KEY (circuitName) REFERENCES Circuit_2(circuitName)
            ON DELETE NO ACTION
);
```

8. GrandPrix_Ref(**circuitName: varchar**, city: varchar)

```
CREATE TABLE GrandPrix_2 (
      circuitName             varchar,          PRIMARY KEY,
      city                    varchar           NOT NULL,
      FOREIGN KEY (circuitName) REFERENCES GrandPrix_Ref(circuitName)
            ON DELETE NO ACTION
);
```

9. GrandPrix_3(**circuitName: varchar**, country: varchar)

```
CREATE TABLE GrandPrix_3 (
      circuitName             varchar           PRIMARY KEY,
      country                 varchar           NOT NULL,
      FOREIGN KEY (circuitName) REFERENCES GrandPrix_Ref(circuitName)
            ON DELETE NO ACTION
);
```

10. GrandPrix_4(**year: year**, **circuitName: varchar**, attendance: int)

```
CREATE TABLE GrandPrix_4 (
      year                    year,
      circuitName             varchar,
      attendance              int,
      PRIMARY KEY (year, circuitName),
      FOREIGN KEY (year, circuitName) REFERENCES GrandPrix_2(year, circuitName)
            ON DELETE NO ACTION
);
```

11. GrandPrix_5(**year: year**, gpName: varchar, **circuitName: varchar**)

```
CREATE TABLE GrandPrix_5 (
      year                    year,
      gpName                  varchar,
      circuitName             varchar           NOT NULL,
      PRIMARY KEY (year, gpName),
      FOREIGN KEY (year, circuitName) REFERENCES GrandPrix_2(year, circuitName)
            ON DELETE NO ACTION
);
```

12. Circuit_Ref(#OfLaps, length: integer)

```
CREATE TABLE Circuit_Ref (
      #OfLaps                 int               PRIMARY KEY,
      length                  int
);
```

13. Circuit_2(circuitName: varchar, **#OfLaps**: **integer,** type: varchar)

```
CREATE TABLE Circuit_2 (
      circuitName             varchar           PRIMARY KEY,
      #OfLaps                 int,
      type                    varchar,
      FOREIGN KEY (#OfLaps) REFERENCES Circuit_Ref(#OfLaps)
            ON DELETE CASCADE
);
```

14. GrandPrix_ConstructorStanding_Ref(position: integer, points: integer)

```
CREATE TABLE GrandPrix_ConstructorStanding_Ref (
      position                int               PRIMARY KEY,
      points                  int               DEFAULT 0 NOT NULL
);
```

15. GrandPrix_ConstructorStanding_2(**position: integer**, **gpName: varchar**, **year: year**)

```
CREATE TABLE GrandPrix_ConstructorStanding_2 (
      position                int,
      gpName                  varchar,
      year                    year,
      PRIMARY KEY (position, gpName, year),
      FOREIGN KEY (gpName, year) REFERENCES GrandPrix_5(gpName, year)
```

```
            ON DELETE NO ACTION,
      FOREIGN KEY (position) REFERENCES GrandPrix_ConstructorStanding_Ref(position)
            ON DELETE NO ACTION
);
```

16. GrandPrix_DriverStanding_Ref(racePosition: integer, points: integer)

```
CREATE TABLE GrandPrix_DriverStanding_Ref (
      racePosition            int                 PRIMARY KEY,
      points                  int                 DEFAULT 0 NOT NULL
);
```

17. GrandPrix_DriverStanding_2(**racePosition: integer**, **gpName: varchar**, **year: year**, qualifyingPosition: integer)

```
CREATE TABLE GrandPrix_DriverStanding_2 (
      racePosition            int,
      gpName                  varchar,
      year                    year,
      qualifyingPosition      int,
      PRIMARY KEY (racePosition, gpName, year),
      FOREIGN KEY (gpName, year) REFERENCES GrandPrix_5(gpName, year)
            ON DELETE NO ACTION,
      FOREIGN KEY (racePosition) REFERENCES
            GrandPrix_DriverStanding_Ref(racePosition)
            ON DELETE NO ACTION
);
```

18. Driver(**employeeId: integer**, #OfPodiums: integer, #OfWins: integer, driverNumber: integer, #OfPolePositions: integer)

```
CREATE TABLE Driver (
      employeeId              int                 PRIMARY KEY,
      #OfPodiums              int                 DEFAULT 0 NOT NULL,
      #OfWins                 int                 DEFAULT 0 NOT NULL,
      driverNumber            int                 NOT NULL,
      #OfPolePositions        int                 DEFAULT 0 NOT NULL,
      FOREIGN KEY (employeeId) REFERENCES TeamMember(employeeId)
            ON DELETE NO ACTION
);
```

19. Sponsors(**companyName: varchar**, **constructorName: varchar,** sponsorshipAmount:integer)

```
CREATE TABLE Sponsors (
      companyName             varchar,
      constructorName         varcar,
      sponsorshipAmount       int,
      PRIMARY KEY (companyName, constructorName),
      FOREIGN KEY (companyName) REFERENCES Sponsor(companyName)
            ON DELETE NO ACTION,
      FOREIGN KEY (constructorName) REFERENCES Constructor(constructorName)
```

```
            ON DELETE NO ACTION
);
```

20. WorksWith(**constructorName: varchar, employeeId: integer**, since: date)

```
CREATE TABLE WorksWith (
      constructorName         varchar,
      employeeId              int,
      since                   date,
      PRIMARY KEY (constructorName, employeeId),
      FOREIGN KEY (constructorName) REFERENCES Constructor(constructorName)
            ON DELETE CASCADE,
      FOREIGN KEY (employeeId) REFERENCES TeamMember(employeeID)
            ON DELETE CASCADE
);
```

21. Drives(**model: varchar, employeeId: integer**)

```
CREATE TABLE Drives (
      model                   varchar,
      employeeId              int,
      PRIMARY KEY (model, employeeId),
      FOREIGN KEY (model) REFERENCES Car(model)
            ON DELETE CASCADE,
      FOREIGN KEY (employeeId) REFERENCES Driver(employeeId)
            ON DELETE CASCADE
);
```

22. InRelationshipWith(**partnerId: integer, employeeId: integer**, since: date, instagramHandle: varchar, instagramFollowers: integer)

```
CREATE TABLE InRelationshipWith (
      partnerId               int,
      employeeId              int,
      since                   date,
      instagramHandle         varchar,
      instagramFollowers      int,
      PRIMARY KEY (partnerId, employeeId),
      FOREIGN KEY (partnerId) REFERENCES Partner_Ref(partnerId)
            ON DELETE CASCADE,
      FOREIGN KEY (employeeId) REFERENCES Driver(employeeId)
            ON DELETE CASCADE
);
```

23. ConstructorHolds(**position: integer, gpName: varchar, year: year, constructorName: varchar**)

***Note**: since the ConstructorHolds relationship references the weak entity ConstructorStanding, its primary keys are also foreign keys from GrandPrix_ConstructorStanding and its parent entity GrandPrix. We have decided to attribute all the foreign key references to the weak entity GrandPrix_ConstructorStanding as an intuitive design choice.*

```
CREATE TABLE ConstructorHolds (
      position                int,
      gpName                  varchar,
      year                    year,
      constructorName         varchar,
      PRIMARY KEY (position, gpName, year, constructorName),
      FOREIGN KEY (gpName, year, position) REFERENCES
            GrandPrix_ConstructorStanding(gpName, year, position)
            ON DELETE CASCADE,
      FOREIGN KEY (constructorName) REFERENCES Constructor(constructorName)
            ON DELETE CASCADE
);
```

24. DriverHolds(<u>**racePosition: integer, gpName: varchar, year: year, employeeID: integer**</u>)

    Note: *since the DriverHolds relationship references the weak entity DriverStanding, its primary keys are also foreign keys from GrandPrix_DriverStanding and its parent entity GrandPrix. We have decided to attribute all the foreign key references to the weak entity GrandPrix_DriverStanding as an intuitive design choice.*

```
CREATE TABLE DriverHolds (
      racePosition            int,
      gpName                  varchar,
      year                    year,
      employeeId              varchar,
      PRIMARY KEY (racePosition, gpName, year, employeeId),
      FOREIGN KEY (gpName, year, racePosition) REFERENCES
            GrandPrix_DriverStanding(gpName, year, racePosition)
            ON DELETE CASCADE,
      FOREIGN KEY (employeeId) REFERENCES Driver(employeeId)
            ON DELETE CASCADE
);
```

INSERT Statements:

1. Sponsor(<u>CompanyName: varchar</u>, Industry: varchar)

```
INSERT INTO Sponsor (companyName, industry)
VALUES (‘Oracle’, ‘Tech’),
       (‘Zoom’, ‘Tech’),
       (‘Tommy Hilfiger’, ‘Apparel’),
       (‘Monster Energy’, ‘Beverage’),
       (‘Ray-Ban’, ‘Apparel’),
       (‘Shell’, ‘Oil and Gas’),
       (‘Chrome’, ‘Tech’),
       (‘Michelob Ultra’, ‘Beverage’),
       (‘Tik Tok’, ‘Tech’);
```

2. Constructor(ConstructorName: varchar, Nationality: varchar, #OfWins: integer)

```
INSERT INTO Constructor (constructorName, nationality, #OfWins),
VALUES (‘Red Bull Racing’, ‘Austria’, ‘6’),
       (‘Mercedes’, ‘Germany’, ‘8’),
       (‘Ferrari’, ‘Italy’, ‘15’),
       (‘Aston Martin’, ‘England’, ‘0’),
       (‘McLaren’, ‘England’, ‘20’),
       (‘Alpine Renault’, ‘France’, ‘0’),
       (‘Williams’, ‘England’, ‘9’),
       (‘Alfa Romeo’, ‘Italy’, ‘5’),
       (‘Haas’, ‘United States’, ‘0’),
       (‘AlphaTauri’, ‘Italy’, ‘0’);
```

3. TeamMember(EmployeeId: integer, FirstName: varchar, LastName: varchar, Nationality: varchar, DateOfBirth: date, Salary: integer, Job: varchar)

```
INSERT INTO TeamMember (employeeId, firstName, lastName, nationality, dateOfBirth,
salary, job)
VALUES (‘1’, ‘Toto’, ‘Wolff’, ‘Austria’, ‘1972-01-12’, ‘26000000’, ‘Team
Principal’),
       (‘2’, ‘Christian’, ‘Horner’, ‘England’, ‘1973-11-16’, ‘10000000’, ‘Team
Principal’),
       (‘3’, ‘Zak’, ‘Brown’, ‘United States’, ‘1971-11-07’, ‘5000000’, ‘CEO’),
       (‘4’, ‘Hannah’, ‘Schmitz’, ‘England’, ‘1985-05-01’, ‘154000’, ‘Principal
Strategy Engineer’),
       (‘5’, ‘Peter’, ‘Bonnington’, ‘England’, ‘1975-02-12’, ‘450000’, ‘Senior Race
Engineer’)
       (‘6’, ‘Oscar’, ‘Piastri’, ‘2001-04-06’, ‘2000000’, ‘Driver’),
       (‘7’, ‘Logan’, ‘Sargeant’, ‘2000-12-31’, ‘1000000’, ‘Driver’),
       (‘8’, ‘Yuki’, ‘Tsunoda’, ‘2000–05-11’, ‘1000000’, ‘Driver’),
       (‘9’, ‘Lando’, ‘Norris’, ‘1999-11-13’, ‘20000000’, ‘Driver’),
       (‘10’, ‘Zhou’, ‘Guanyu’, ‘1999-05-30’, ‘2000000’, ‘Driver’),
       (‘11’, ‘Lance’, ‘Stroll’, ‘1998-10-29’, ‘2800000’, ‘Driver’),
       (‘12’, ‘George’, ‘Russell’, ‘1998-02-15’, ‘8000000’, ‘Driver’),
       (‘13’, ‘Charles’, ‘Leclerc’, ‘1997-10-16’, ‘24000000’, ‘Driver’),
       (‘14’, ‘Esteban’, ‘Ocon’, ‘1996-09-17’, ‘6000000’, ‘Driver’),
       (‘15’, ‘Alex’, ‘Albon’, ‘1996-03-23’, ‘3000000’, ‘Driver’),
       (‘16’, ‘Pierre’, ‘Gasly’, ‘1996-02-07’, ‘5000000’, ‘Driver’),
       (‘17’, ‘Carlos’, ‘Sainz’, ‘1994-09-01’, ‘12000000’, ‘Driver’),
       (‘18’, ‘Kevin’, ‘Magnussen’, ‘1992-10-05’, ‘5000000’, ‘Driver’),
       (‘19’, ‘Sergio’, ‘Perez’, ‘1990-01-26’, ‘10000000’, ‘Driver’),
       (‘20’, ‘Valtteri’, ‘Bottas’, ‘1989-08-28’, ‘10000000’, ‘Driver’),
       (‘21’, ‘Nico’, ‘Hulkenberg’, ‘1987-08-19’, ‘2000000’, ‘Driver’),
       (‘22’, ‘Lewis’, ‘Hamilton’, ‘1985-01-07’, ‘37000000’, ‘Driver’),
       (‘23’, ‘Max’, ‘Verstappen’, ‘1997-09-30’, ‘50000000’, ‘Driver’),
       (‘24’, ‘Fernando’, ‘Alonso’, ‘1981-07-29’, ‘20000000’, ‘Driver’);
```

4. Car(model: varchar, engine: varchar, **constructorName: varchar**)

```
INSERT INTO Car (model, engine, constructorName)
VALUES (‘RB19’, ‘Red Bull Powertrains - Honda’, ‘Red Bull’),
       (‘SF-23’ ‘Ferrari’, ‘Ferrari’),
       (‘W14’, ‘Mercedes’, ‘Mercedes’),
       (‘A523’, ‘Renault’, ‘Alpine’),
       (‘MCL60’, ‘Mercedes’, ‘McLaren’),
       (‘C43’, ‘Ferrari’, ‘Alfa Romeo’),
       (‘AMR23’, ‘Mercedes’, ‘Aston Martin’),
       (‘VF-23’, ‘Ferrari’, ‘Haas’),
       (‘AT04’, ‘Red Bull Powertrains - Honda’, ‘AlphaTauri’),
       (‘FW45’, ‘Mercedes’, ‘Williams’);
```

5. Partner_Ref(partnerID: integer, partnerName: varchar, instagramHandle: varchar)

```
INSERT INTO Partner_Ref (partnerId, partnerName, instagramHandle)
VALUES (‘1’, ‘Kelly Piquet’, ‘kellypiquet’),
       (‘2’, ‘Kika Gomes’, ‘francisca.cgomes’),
       (‘3’, ‘Carmen Montero Mundt’, ‘carmenmmundt’),
       (‘4’, ‘Lily Zneimer’, ’lilyzneimer’),
       (‘5’, ‘Tiffany Cromwell’, ‘tiffanycromwell’),
       (‘6’, ‘Lily Muni He’, ‘lilymhe’);
```

6. Partner_2(**instagramHandle: varchar**, instagramFollowers: integer)

```
INSERT INTO Partner_2 (instagramHandle, instagramFollowers)
```

```
VALUES (‘kellypiquet’, ‘1300000’),
       (‘francisca.cgomes’, ‘537000’),
       (‘carmenmmundt’, ‘309000’),
       (’lilyzneimer’, ‘800’),
       (‘tiffanycromwell’, ‘195000’),
       (‘lilymhe’, ‘688000’);
```

7. GrandPrix_Ref(year: year, **circuitName: varchar,** viewership: integer)

```
INSERT INTO GrandPrix_Ref (year, circuitName, viewership)
VALUES (‘2023’, ‘Bahrain International Circuit’, ‘1300000’),
       (‘2023’, ‘Albert Park Circuit’, ‘2950000’),
       (‘2023’, ‘Monaco’, ‘1790000’),
       (‘2023’, ‘Silverstone Circuit’, ‘2350000’),
       (‘2023’, ‘Marina Bay Street Circuit’, ‘1300000’);
```

8. GrandPrix_2(**circuitName: varchar**, city: varchar)

```
INSERT INTO GrandPrix_2 (circuitName, city)
VALUES (‘Bahrain International Circuit’, ‘Sakhir’),
       (‘Albert Park Circuit’, ‘Melbourne’),
       (‘Monaco’, ‘Monte Carlo’),
       (‘Silverstone Circuit’, ‘Towcester’),
       (‘Marina Bay Street Circuit’, ‘Marina Bay’);
```

9. GrandPrix_3(**circuitName: varchar**, country: varchar)

```
INSERT INTO GrandPrix_3 (circuitName, country)
VALUES (‘Bahrain International Circuit’, ‘Bahrain’),
       (‘Albert Park Circuit’, ‘Australia’),
       (‘Monaco’, ‘Monaco’),
       (‘Silverstone Circuit’, ‘England’),
       (‘Marina Bay Street Circuit’, ‘Singapore’);
```

10. GrandPrix_4(**year: year**, **circuitName: varchar**, attendance: int)

```
INSERT INTO GrandPrix_4 (year, circuitName, attendance)
VALUES (‘2023’, ‘Bahrain International Circuit’, ‘36000’),
       (‘2023’, ‘Albert Park Circuit’, ‘444600’),
       (‘2023’, ‘Monaco’, ‘200000’),
       (‘2023’, ‘Silverstone Circuit’, ‘480000’),
       (‘2023’, ‘Marina Bay Street Circuit’, ‘264000’);
```

11. GrandPrix_5(**year: year,** gpName: varchar, **circuitName: varchar**)

```
INSERT INTO GrandPrix_5 (year, gpName, circuitName)
VALUES (‘2023’, ‘Bahrain Grand Prix’, ‘Bahrain International Circuit’),
       (‘2023’, ‘Australian Grand Prix’, Albert Park Circuit’),
       (‘2023’, ‘Monaco Grand Prix’, ‘Monaco’),
       (‘2023’, ‘British Grand Prix’, ‘Silverstone Circuit’),
       (‘2023’, ‘Singapore Grand Prix’, ‘Marina Bay Street Circuit’);
```

12. Circuit_Ref(#Oflaps, length: integer)

```
INSERT INTO Circuit_Ref (#OfLaps, length)
VALUES (‘57’, ‘308’),
       (‘58’, ‘307’),
       (‘78’, ‘260’),
       (‘52’, ‘306’),
       (‘61’, ‘308’);
```

13. Circuit_2(circuitName: varchar, **#OfLaps**: **integer,** type: varchar)

```
INSERT INTO Circuit_2 (circuitName, #OfLaps, type)
VALUES (‘Bahrain International Circuit’, ‘57’, ‘race’),
       (‘Albert Park Circuit’, ‘58’, ‘street’),
       (‘Monaco’, ‘78’, ‘street’),
       (‘Silverstone Circuit’, ‘52’, ‘race’),
       (‘Marina Bay Street Circuit’, ‘61’, ‘street’);
```

14. GrandPrix_ConstructorStanding_Ref(position: integer, points: integer)

```
INSERT INTO GrandPrix_ConstructorStanding_Ref (position, points)
VALUES (‘1’, ‘37’),
       (‘2’, ‘30’),
       (‘3’, ‘25’),
       (‘1’, ‘25’),
       (‘2’, ‘27’);
```

15. GrandPrix_ConstructorStanding_2(**position: integer**, **gpName: varchar, year: year**)

```
INSERT INTO GrandPrix_ConstructorStanding_2 (position, gpName, year)
VALUES (‘1’, ‘Singapore Grand Prix’, ‘2023’),
       (‘2’, ‘British Grand Prix’, ‘2023’),
       (‘3’, ‘British Grand Prix’, ‘2023’),
       (‘1’, ‘Monaco Grand Prix’, ‘2023’),
       (‘2’, ‘Australian Grand Prix’, ‘2023’);
```

16. GrandPrix_DriverStanding_Ref(racePosition: integer, points: integer)

```
INSERT INTO GrandPrix_DriverStanding_Ref (racePosition, points)
VALUES (‘1’, ‘25’),
       (‘4’, ‘12’),
       (‘2’, ‘18’),
       (‘4’, ‘12’),
       (‘3’, ‘15’),
       (‘4’, ‘10’),
       (‘1’, ‘25’),
       (‘16’, ‘0’),
       (‘3’, ‘15’),
       (‘4’, ‘12’);
```

17. GrandPrix_DriverStanding_2(**racePosition: integer**, **gpName: varchar, year: integer**, qualifyingPosition: integer)

```
INSERT INTO GrandPrix_DriverStanding_2 (racePosition, gpName, year,
qualifyingPosition)
VALUES (‘1’, ‘Singapore Grand Prix’, ‘2023’, ‘1’),
       (‘4’, ‘Singapore Grand Prix’, ‘2023’, 3’),
       (‘2’, ‘British Grand Prix’, ‘2023’, ‘2’),
       (‘4’, ‘British Grand Prix’, ‘2023’, ‘3’),
       (‘3’, ‘British Grand Prix’, ‘2023’, ‘6’),
       (‘4’, ‘British Grand Prix’, ‘2023’, ‘7’),
       (‘1’, ‘Monaco Grand Prix’, ‘2023’, ‘1’),
       (‘16’, ‘Monaco Grand Prix’, ‘2023’, ‘20’),
       (‘3’, ‘Australian Grand Prix’, ‘2023’, ‘4’),
       (‘4’, ‘Australian Grand Prix’, ‘2023’, ‘6’);
```

18. Driver(**employeeId: integer**, #OfPodiums: integer, #OfWins: integer, driverNumber: integer, #OfPolePositions: integer)

```
INSERT INTO Driver (employeeId, #OfPodiums, #OfWins, driverNumber, #OfPolePositions)
VALUES (‘6’, ‘2’, ‘0’, ‘81’, ‘1’),
       (‘7’, ‘0’, ‘0’, ‘2’, ‘0’),
       (‘8’, ‘0’, ‘0’, ‘22’, ‘0’),
       (‘9’, ‘11’, ‘0’, ‘4’, ‘1’),
       (‘10’, ‘0’, ‘0’, ‘24’, ‘0’),
       (‘11’, ‘3’, ‘0’, ‘18’, ‘1’),
       (‘12’, ‘10’, ‘1’, ‘63’, ‘1’),
       (‘13’, ‘27’, ‘5’, ‘16’, ‘19’),
       (‘14’, ‘3’, ‘1’, ‘31’, ‘0’),
       (‘15’, ‘2’, ‘0’, ‘23’, ‘0’),
       (‘16’, ‘4’, ‘1’, ‘10’, ‘0’),
       (‘17’, ‘17’, ‘2’, ‘55’, ‘5’),
       (‘18’, ‘1’, ‘0’, ‘20’, ‘1’),
       (‘19’, ‘34’, ‘6’, ‘11’, ‘3’),
       (‘20’, ‘67’, ‘10’, ‘77’, ‘20’),
       (‘21’, ‘0’, ‘0’, ‘27’, ‘1’),
       (‘22’, ‘196’, ‘103’, ‘44’, ‘104’),
       (‘23’, ‘93’, ‘49’, ‘1’, ‘30’),
       (‘24’, ‘105’, ‘32’, ‘14’, ‘22’);
```

19. Sponsors(**companyName: varchar, constructorName: varchar**, sponsorshipAmount:integer)

```
INSERT INTO Sponsors (companyName, constructorName, sponsorshipAmount)
VALUES (‘Oracle’, ‘Red Bull Racing’, ‘500000000’),
       (‘Zoom’, ‘Red Bull Racing’, ‘150000000’),
       (‘Tommy Hilfiger’, ‘Mercedes’, ‘50000000’),
       (‘Monster Energy’, ‘Mercedes’, ‘85000000’),
       (‘Ray-Ban’, ‘Ferrari’, ‘290000000’),
       (‘Shell’, ‘Ferrari’, ‘350000000’),
       (‘Chrome’, ‘McLaren’, ‘420000000’),
       (‘Michelob Ultra’, ‘Williams’, ‘41000000’),
       (‘Tik Tok’, ‘Aston Martin’, NULL);
```

20. WorksWith(**constructorName: varchar, employeeId: integer**, since: date)

```
INSERT INTO WorksWith (constructorName, employeeId, since)
VALUES (‘Red Bull Racing’, ‘2’, ‘2005-01-01’),
       (‘Mercedes’, ‘1’, ‘2013-01-01’),
       (‘McLaren’, ‘3’, ‘2016-11-01’),
       (‘Red Bull Racing’, ‘4’, ‘2009-11-01’),
       (‘Mercedes’, ‘5’, ‘2011-09-01’),
       (‘McLaren’, ‘6’, ‘2023-03-05’),
       (‘Williams’, ‘7’, ‘2023-03-05’),
       (‘AlphaTauri’, ‘8’, ‘2021-03-28’),
       (‘McLaren’, ‘9’, ‘2019-03-17’),
       (‘Alfa Romeo’, ‘10’, 2022-03-20’),
       (‘Aston Martin’, ‘11’, ‘2017-03-26’),
       (‘Mercedes’, ‘12’, ‘2019-03-17’),
       (‘Ferrari’, ‘13’, ‘2018-03-25’),
       (‘Alpine’, ‘14’, ‘2016-08-28’),
       (‘Williams’, ‘15’, ‘2019-03-17’),
       (‘Alpine’, ‘16’, ‘2017-10-01’),
       (‘Ferrari’, ‘17’, ‘2015-03-15’),
       (‘Haas’, ‘18’, ‘2014-03-16’),
       (‘Red Bull Racing’, ‘19’, ‘2011-03-27’),
       (‘Alfa Romeo’, ‘20’, ‘2013-03-17’),
       (‘Haas’, ‘21’, ‘2010-03-14’),
       (‘Mercedes’, ‘22’, ‘2007-03-28’),
       (‘Red Bull Racing’, ‘23’, ‘2015-03-15’),
       (‘Aston Martin’, ‘24’, ‘2001-03-04’);
```

21. Drives(**model: varchar, employeeId: integer**)

```
INSERT INTO Drives (model, employeeId)
VALUES (‘W14’, ‘1’),
       (‘W14’, ‘22’),
       (‘RB19’, ‘19’),
       (‘RB19’, ‘23’),
       (‘SF-23’, ‘17’),
       (‘SF-23’, ‘13’),
       (‘A523’, ‘14’),
       (‘A523’, ‘16’),
       (‘MCL60’, ‘3’),
       (‘MCL60’, ‘6’),
       (‘C43’, ‘10’),
       (‘C43’, ‘20’),
       (‘AMR23’, ‘24’),
       (‘AMR23’, ‘11’),
       (‘WF-23’, ‘18’),
       (‘WF-23’, ‘21’),
       (‘AT04’, ‘8’),
       (‘FW45’, ‘15’),
       (‘FW45’, ‘7’);
```

22. InRelationshipWith(**partnerId: integer, employeeId: integer**, since: date, instagramHandle: varchar, instagramFollowers: integer)

```
INSERT INTO InRelationshipWith (partnerID, employeeId, since, instagramHandle,
instagramFollowers)
VALUES (‘1’, ‘23’, ‘2020-03-05’, ‘kellypiquet’, ‘1300000’),
       (‘2’, ‘16’, ‘2022-09-13’, ‘francisca.cgomes’, ‘537000’),
       (‘3’, ‘12’, ‘2020-06-20’, ‘carmenmmundt’, ‘309000’),
       (‘4’, ‘6’, ‘2019-01-16’, ‘lilyzneimer’, ‘880’),
       (‘5’, ‘20’, ‘2020-03-29’, ‘tiffanycromwell’, ‘195000’),
       (‘6’, ‘15’, ‘2019-04-12’, ‘lilymhe’, ‘688000’);
```

23. ConstructorHolds(**position: integer, gpName: varchar, year: integer, constructorName: varchar**)

```
INSERT INTO ConstructorHolds (position, gpName, year, constructorName)
VALUES (‘1’, ‘Singapore Grand Prix’, ‘2023’, ‘Ferrari’),
       (‘2’, ‘British Grand Prix’, ‘2023’, ‘McLaren’),
       (‘3’, ‘British Grand Prix’, ‘2023’, ‘Mercedes’),
       (‘1’, ‘Monaco Grand Prix’, ‘2023’, ‘Red Bull Racing’),
       (‘2’, ‘Aston Martin’, ‘Australian Grand Prix’, ‘2023’, ‘Aston Martin’);
```

24. DriverHolds(**racePosition: integer, gpName: varchar, year: integer, employeeID: integer**)

```
INSERT INTO DriverHolds (racePosition, gpName, year, employeeId)
```

```
VALUES (‘1’, ‘Singapore Grand Prix’, ‘2023’, ‘17’),
       (‘4’, ‘Singapore Grand Prix’, ‘2023’, ‘13’),
       (‘2’, ‘British Grand Prix’, ‘2023’, ‘9’),
       (‘4’, ‘British Grand Prix’, ‘2023’, ‘6’),
       (‘3’, ‘British Grand Prix’, ‘2023’, ‘22’),
       (‘4’, ‘British Grand Prix’, ‘2023’, ‘12’),
       (‘1’, ‘Monaco Grand Prix’, ‘2023’, ‘23’),
       (‘16’, ‘Monaco Grand Prix’, ‘2023’, ‘19’),
       (‘3’, ‘Australian Grand Prix’, ‘2023’, ‘24’),
       (‘4’, ‘Australian Grand Prix’, ‘2023’, ‘11’);
```

Milestone 3

University of British Columbia, Vancouver
Department of Computer Science
CPSC 304 Project - The Formula for Success
Milestone #: 3
Date: October 30, 2023
Group Number: 51

Name	Student Number	CS Alias	Email Address
Kira Swinth	38122560	i9b6u	kira.swinth@gmail.com
Kelly Zhang	46990602	b8g1d	kellyz02@student.ubc.ca
Faraneh Yahyaei-Moayyed	60431905	x3y9g	faranehyahyaei@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your email address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Summary:

The domain of our application is sports data management. More specifically, our application focuses on Formula One (F1), a form of international motorsport. The application will store statistics and results for fans to reference and explore.

Timeline:

Database Tasks: *finished by Nov. 15*

- Single, runnable SQL script - *broken down, but to be combined at the end:*
  - □ Faraneh: DDL statements #1-8, INSERT statements #1-8
  - □ Kira: DDL statements #9-16, INSERT statements #9-16
  - □ Kelly: DDL statements #17-24, INSERT statements #17-24

Front End Tasks: *finished by Nov. 22*

- Administrator login (complete with username + password): - *everyone*
  - □ Administrator page display - *everyone*
- Main page display:
  - □ Search bar + "Search by" drop-down display - *everyone*
  - □ Standardized additional filters display + format
    - ■ Faraneh: Season, Drivers filters
    - ■ Kira: Grand Prix, Constructors filters
    - ■ Kelly: Sponsor, Circuit filters
- Query output display - *everyone*

Back End Tasks: *finished by Nov. 22*

- Administrator page: should include INSERT, DELETE, and UPDATE query functionalities - *everyone*
- Main page: search bar functionality + "search by" drop down - *everyone*
- Additional filters specific to the "search by" category that fulfills our required queries:
  ****Keep in Mind: the required queries when making our additional search filters/functionalities (division, nested aggregation with group by, selection, projection, aggregation with having, join, aggregation with group by)*
  - □ Faraneh: Season, Drivers
    - ■ In Season: possibly incorporate Constructor Standing and Driver Standing data
  - □ Kira: Grand Prix, Constructors
    - ■ In Constructors: possibly incorporate Car and Team Member data
  - □ Kelly: Sponsor, Circuit

Project Presentation Tasks: *finished by Dec. 1*

In the PDF File:

- □ Short description of final project - *everyone, already done*
- □ Description of final schema differences from original schema from Milestone 2 - *Faraneh*
- □ Screenshots demonstrating functionality before, during, and after the query using the GUI - *Kira*
- □ Copy of final schema + screenshots - *Kelly*

<u>Check-In Questions:</u>

- Should we use the decomposed relations?
- Clarify workflow steps with the TA - *When exactly is PHP used? How can we integrate frameworks like Bootstrap?*

<u>Front End Queries:</u>

- insert: “add a new driver/constructor/grandprix”
    - Driver
    - Constructor
    - GrandPrix
- delete: “delete a driver/constructor/grandprix”
    - Driver
    - Constructor
    - GrandPrix
- update: “update a driver’s scores”
    - Driver
        - instead of asking for certain attributes, requires all attributes to be inputted
        - searched for via primary key
    - Constructor
- selection:
    - Driver
    - GrandPrix
    - Constructor
    - Circuit
- projection:
- HOw many win’s does a partner’s partner have?

DRIVER

For a given driver (e.g. last name *“driver”*):

- See basic driver attributes

```
SELECT DISTINCT d.numberOfWins, d.numberOfPodiums, d.numberOfPolePositions
FROM Driver d, TeamMember t
WHERE t.lastName = “driver” AND d.employeeId = t.employeeId
```

- See how many wins for a given season (e.g. this year)

```
SELECT COUNT(
FROM DriverHolds dh, Driver d
WHERE dh.year=2023 AND
        dh.position=1 AND
        d.employeeID IN (SELECT d.employeeId
                         FROM Driver d, TeamMember t
                         WHERE t.lastName = “driver” AND d.employeeId = t.employeeId)
```

Interesting Questions (to produce queries)

- For a given Driver
    - How many wins (DriverStanding, RacePosition=1) this season (GrandPrix, Year = 2023)?
    - How many podiums this season?
    - Previous race positions at a given circuit?
    - What circuit type do they perform best at?
    - What car model have they performed best in?
- For a given Constructor
    - How many wins this season?
    - How many podiums this season?
    - What car model have they performed best with?
- For a given Sponsor
    -

Milestone 3 Notes

- Milestone 4 Tasks:
- *Single, runnable* SQL script to create all tables (DDL) + data (INSERT) in database
  - contains assertions + constraints
- PDF File:
  - short description of final project
  - description of final schema vs. schema we turned in (what and why)
  - copy of final schema + screenshots (from SQL Plus) to show what is in each relation after the SQL script (w/ tables + data) has been run
  - list of all SQL queries used + where it can be found in the code (file name + line number)
  - screenshots demonstrating functionality of each query using the GUI, with labels indicating which query it's depicting + whether it was before running the query, during triggering the query using the GUI, and after the query has been run and how the data looks.
    - only for required queries:
      - INSERT
      - CASCADE ON DELETE
      - UPDATE
      - SELECTION
      - PROJECTION
      - JOIN
      - AGGREGATION with GROUP BY
      - AGREGATION with HAVING
      - NESTED AGGREGATION with GROUP BY
      - DIVISION
  - insert
  - update
  -
- README.txt file for anything that you want to add that's not included in the PDF file
-
-
- Set-up Oracle - *everyone*
-
-
- Front-End Tasks:
- Coding required queries: *demonstrate functionality through GUI with screenshots + note down the location of the queries in the code*
  - Faraneh: INSERT, DIVISION, CASCADE ON DELETE
  - Kira: UPDATE, NESTED AGGREGATION with GROUP BY, SELECTION
  - Kelly: PROJECTION, AGGREGATION with HAVING, JOIN, AGGREGATION with GROUP BY

- Preliminary GUI Tasks:
  - Database Manager Login + View:
    - should include restricted access to INSERT + UPDATE + DELETE operations
  - Public View:
    - Search by: Season, Drivers, Constructors, Grand Prix, Circuit, Sponsor
      - GrandPrix display would be a join with all the GrandPrix relations
      - Season would display the rankings
      - Driver would tell us information about the drivers
        - Option to display Partners as well → joining Driver + Partner
    - Search Bar Functionality - *could either be one of:*
      - User types in query that will trigger a list of matching data, otherwise a "No data for this query" message will appear
      - User selects from a list of data relevant to the category that they're querying from
    - Front End Tasks:
      - Main search bar functionality + "search by" drop down - *everyone*
      - Additional filters specific to the "search by" category that fulfills our required queries
        - Faraneh: Season, Drivers
          - In Season: possibly incorporate Constructor Standing and Driver Standing data
        - Kira: Grand Prix, Constructors
          - In Constructors: possibly incorporate Car and Team Member data
        - Kelly: Sponsor, Circuit
        - *keep in mind the required queries when making our additional search filters/functionalities (division, nested aggregation with group by, selection, projection, aggregation with having, join, aggregation with group by)*
      - Displaying the query output - *everyone*
    -
  - Use Bootstrap Framework?
-
- Back-End Tasks:
- Single, runnable SQL script:
  - Faraneh: DDL 1-8, INSERT 1-8
  - Kira: DDL 9-16, INSERT 9-16
  - Kelly: DDL 17-24, INSERT 17-24

Milestone 4

University of British Columbia, Vancouver
Department of Computer Science
CPSC 304 Project - The Formula for Success
Milestone #: 4
Date: December 1, 2023
Group Number: 51

Name	**Student Number**	**CS Alias**	**Email Address**
Kira Swinth	38122560	i9b6u	kira.swinth@gmail.com
Kelly Zhang	46990602	b8g1d	kellyz02@student.ubc.ca
Faraneh Yahyaei-Moayyed	60431905	x3y9g	faranehyahyaei@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your email address, and then let us assign you to a TA for your project supervisor.)

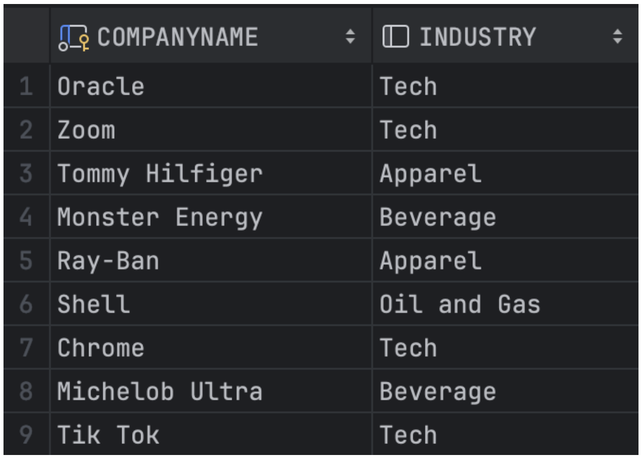
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Description of Project:
Our final application is in the domain of sports data management that focuses on Formula One (F1), a form of international motorsport. The application is able to store statistics and results for fans to reference and explore. Our webapp shows the parts of the database that model important components of the F1 system including Drivers, Constructors (i.e. "teams"), Circuits, Races, Standings and additional ones to accurately represent the intricacies of F1's structure. In addition, the application also shows relationships such as which races drivers compete in, the results of those races, the overall performance of a team in a given season, and many more. Our project focuses on these core components of F1 as the database is intended to support fan engagement and activities, such as F1 betting and fantasy leagues.

Description of Project Schema:
Nothing was changed from the initial schema that we submitted post normalization in milestone 2 and what we ended up implementing in our final application.

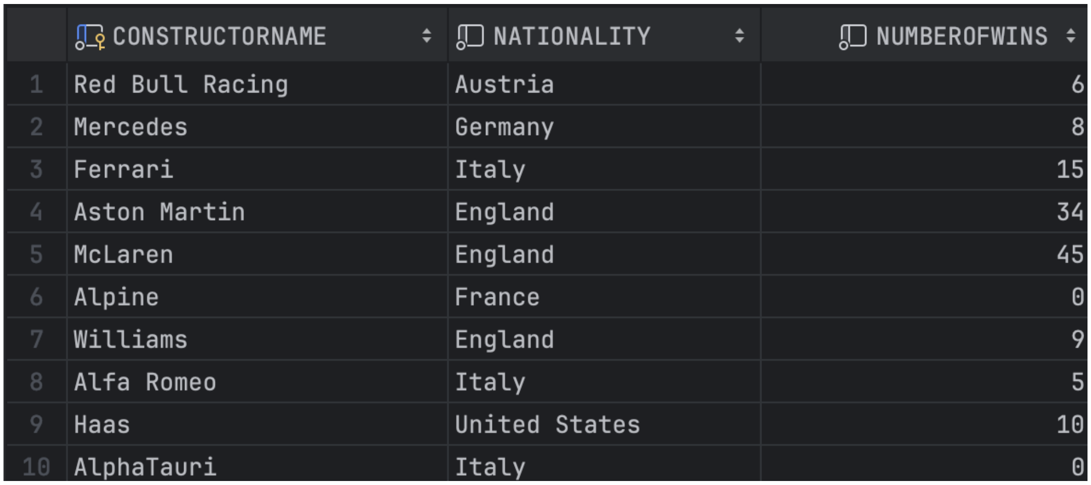
Schema and its Data:

1. Sponsor(companyName, industry)



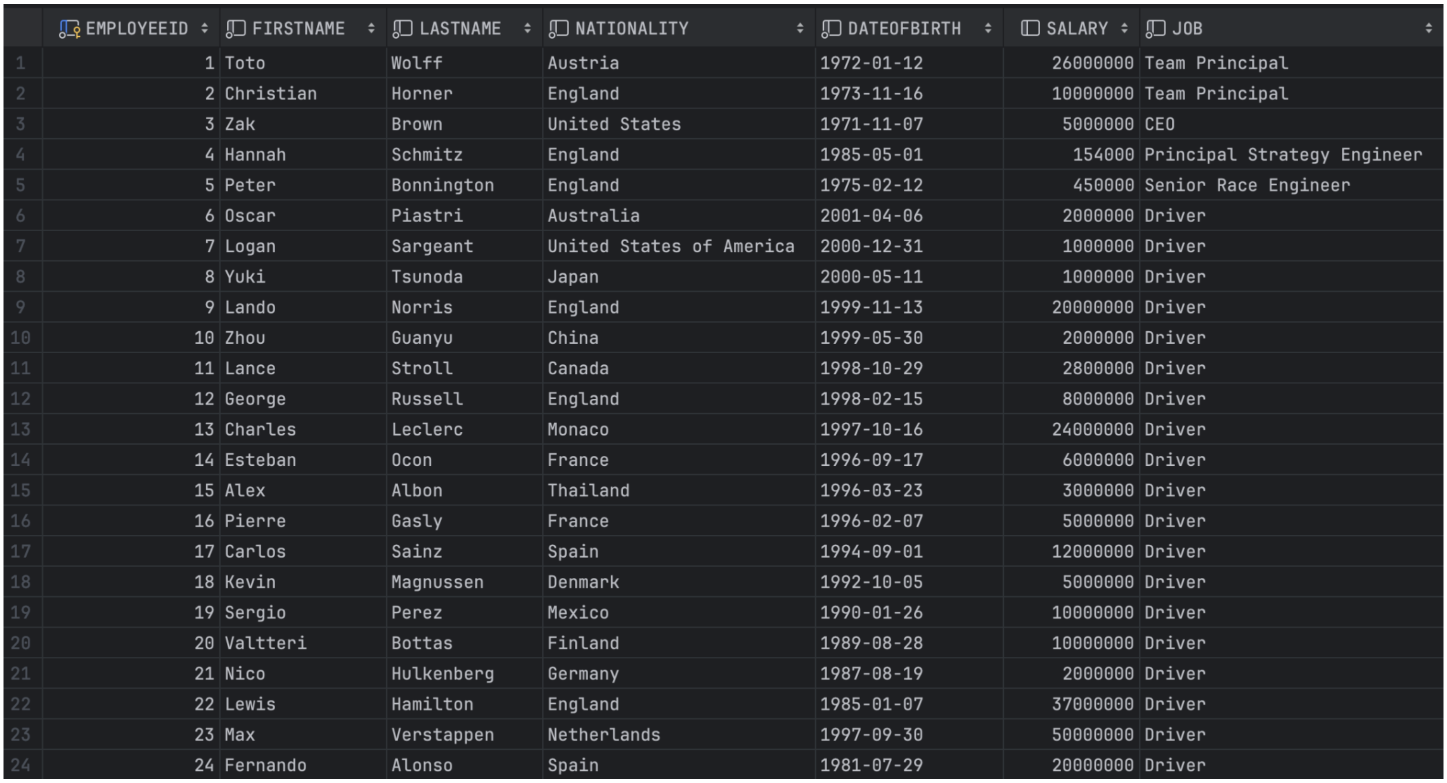
	COMPANYNAME	INDUSTRY
1	Oracle	Tech
2	Zoom	Tech
3	Tommy Hilfiger	Apparel
4	Monster Energy	Beverage
5	Ray-Ban	Apparel
6	Shell	Oil and Gas
7	Chrome	Tech
8	Michelob Ultra	Beverage
9	Tik Tok	Tech

2. Constructor(constructorName, nationality, numberOfWins)



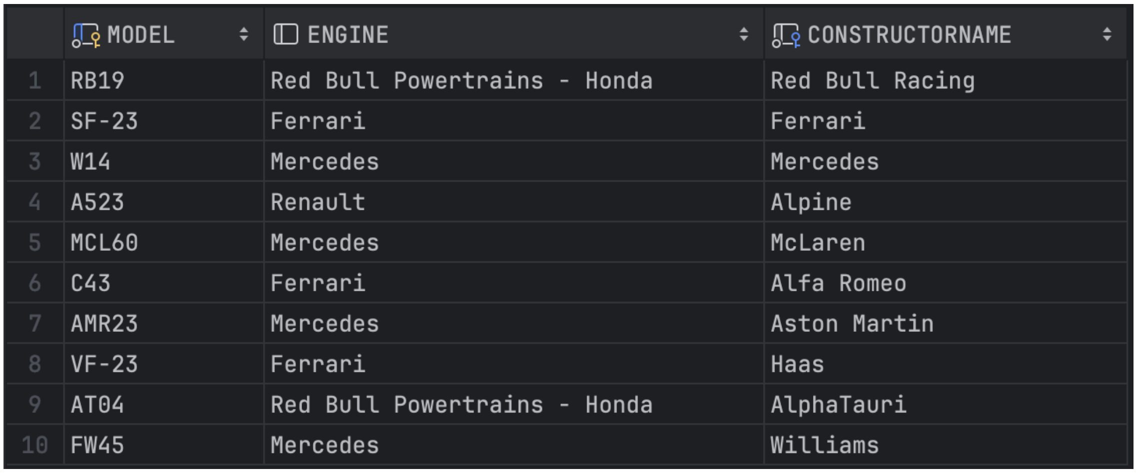
	CONSTRUCTORNAME	NATIONALITY	NUMBEROFWINS
1	Red Bull Racing	Austria	6
2	Mercedes	Germany	8
3	Ferrari	Italy	15
4	Aston Martin	England	34
5	McLaren	England	45
6	Alpine	France	0
7	Williams	England	9
8	Alfa Romeo	Italy	5
9	Haas	United States	10
10	AlphaTauri	Italy	0

3. TeamMember(employeeId, firstName, lastName, nationality,, dateOfBirth, salary, job)



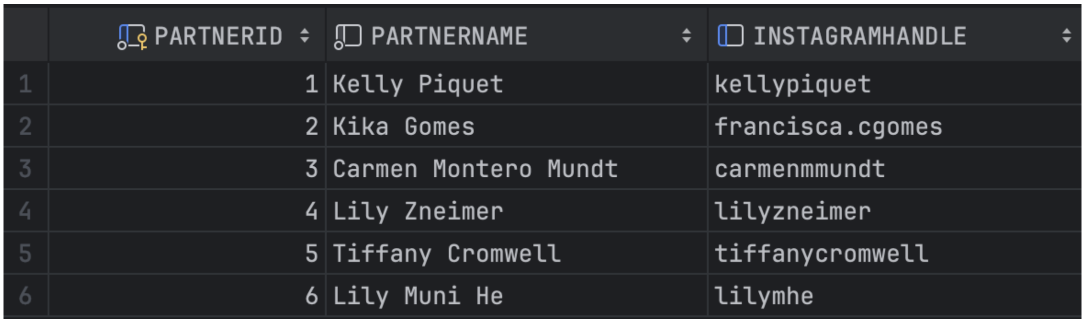
	EMPLOYEEID	FIRSTNAME	LASTNAME	NATIONALITY	DATEOFBIRTH	SALARY	JOB
1	1	Toto	Wolff	Austria	1972-01-12	26000000	Team Principal
2	2	Christian	Horner	England	1973-11-16	10000000	Team Principal
3	3	Zak	Brown	United States	1971-11-07	5000000	CEO
4	4	Hannah	Schmitz	England	1985-05-01	154000	Principal Strategy Engineer
5	5	Peter	Bonnington	England	1975-02-12	450000	Senior Race Engineer
6	6	Oscar	Piastri	Australia	2001-04-06	2000000	Driver
7	7	Logan	Sargeant	United States of America	2000-12-31	1000000	Driver
8	8	Yuki	Tsunoda	Japan	2000-05-11	1000000	Driver
9	9	Lando	Norris	England	1999-11-13	20000000	Driver
10	10	Zhou	Guanyu	China	1999-05-30	2000000	Driver
11	11	Lance	Stroll	Canada	1998-10-29	2800000	Driver
12	12	George	Russell	England	1998-02-15	8000000	Driver
13	13	Charles	Leclerc	Monaco	1997-10-16	24000000	Driver
14	14	Esteban	Ocon	France	1996-09-17	6000000	Driver
15	15	Alex	Albon	Thailand	1996-03-23	3000000	Driver
16	16	Pierre	Gasly	France	1996-02-07	5000000	Driver
17	17	Carlos	Sainz	Spain	1994-09-01	12000000	Driver
18	18	Kevin	Magnussen	Denmark	1992-10-05	5000000	Driver
19	19	Sergio	Perez	Mexico	1990-01-26	10000000	Driver
20	20	Valtteri	Bottas	Finland	1989-08-28	10000000	Driver
21	21	Nico	Hulkenberg	Germany	1987-08-19	2000000	Driver
22	22	Lewis	Hamilton	England	1985-01-07	37000000	Driver
23	23	Max	Verstappen	Netherlands	1997-09-30	50000000	Driver
24	24	Fernando	Alonso	Spain	1981-07-29	20000000	Driver

4. Car(<u>model</u>, engine, **constructorName**)



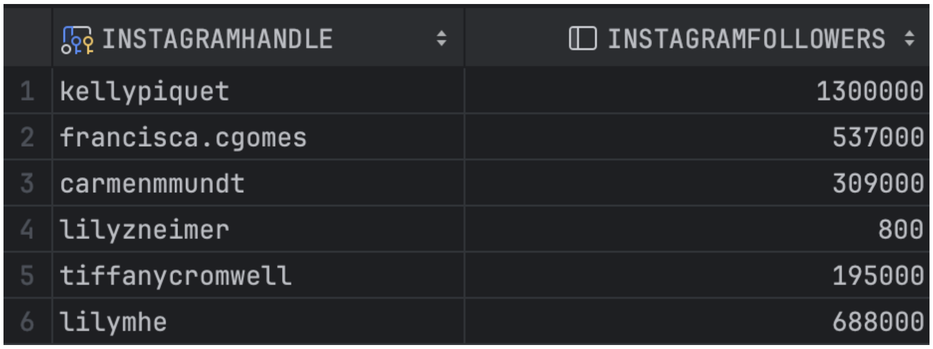
	MODEL	ENGINE	CONSTRUCTORNAME
1	RB19	Red Bull Powertrains - Honda	Red Bull Racing
2	SF-23	Ferrari	Ferrari
3	W14	Mercedes	Mercedes
4	A523	Renault	Alpine
5	MCL60	Mercedes	McLaren
6	C43	Ferrari	Alfa Romeo
7	AMR23	Mercedes	Aston Martin
8	VF-23	Ferrari	Haas
9	AT04	Red Bull Powertrains - Honda	AlphaTauri
10	FW45	Mercedes	Williams

5. Partner_Ref(<u>partnerId</u>, partnerName, instagramHandle).



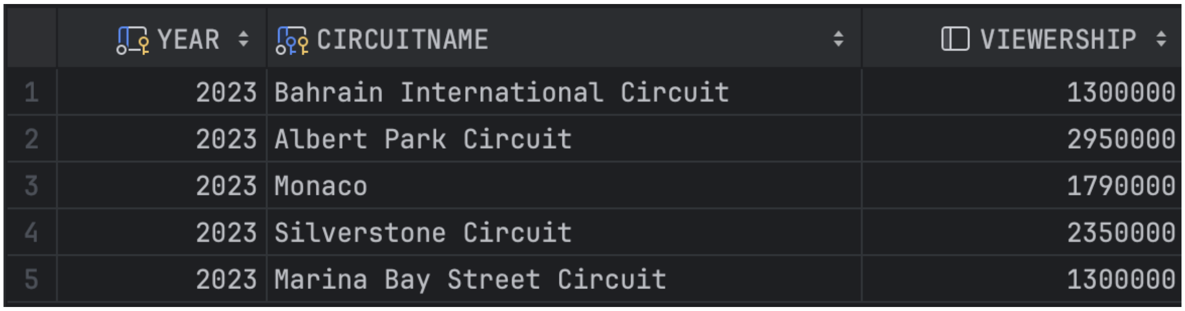
	PARTNERID	PARTNERNAME	INSTAGRAMHANDLE
1	1	Kelly Piquet	kellypiquet
2	2	Kika Gomes	francisca.cgomes
3	3	Carmen Montero Mundt	carmenmmundt
4	4	Lily Zneimer	lilyzneimer
5	5	Tiffany Cromwell	tiffanycromwell
6	6	Lily Muni He	lilymhe

6. Partner_2(<u>**instagramHandle**</u>, instagramFollowers)



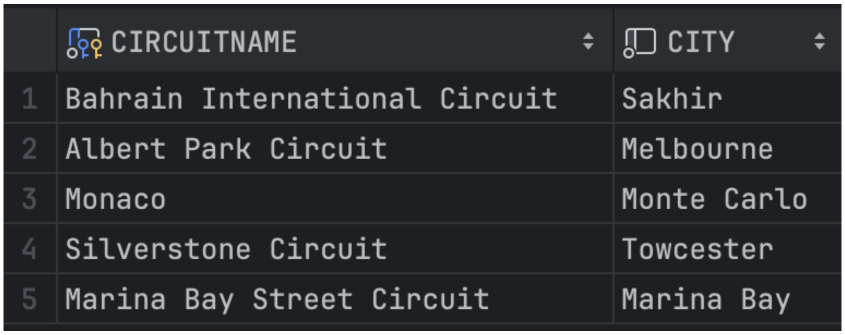
	INSTAGRAMHANDLE	INSTAGRAMFOLLOWERS
1	kellypiquet	1300000
2	francisca.cgomes	537000
3	carmenmmundt	309000
4	lilyzneimer	800
5	tiffanycromwell	195000
6	lilymhe	688000

7. GrandPrix_2(year, **circuitName**, viewership)



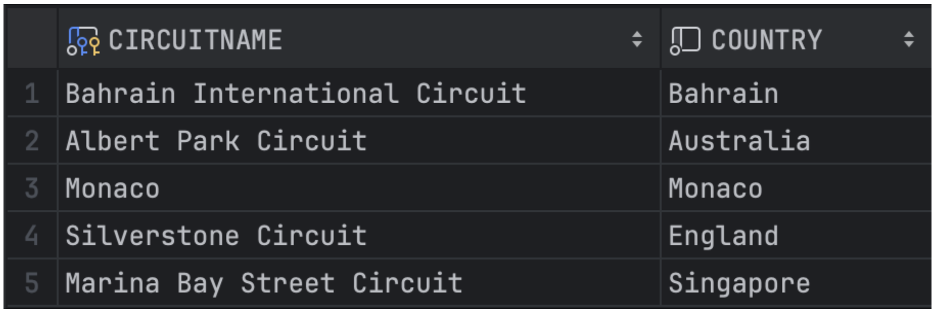
	YEAR	CIRCUITNAME	VIEWERSHIP
1	2023	Bahrain International Circuit	1300000
2	2023	Albert Park Circuit	2950000
3	2023	Monaco	1790000
4	2023	Silverstone Circuit	2350000
5	2023	Marina Bay Street Circuit	1300000

8. GrandPrix_Ref(**circuitName**, city)



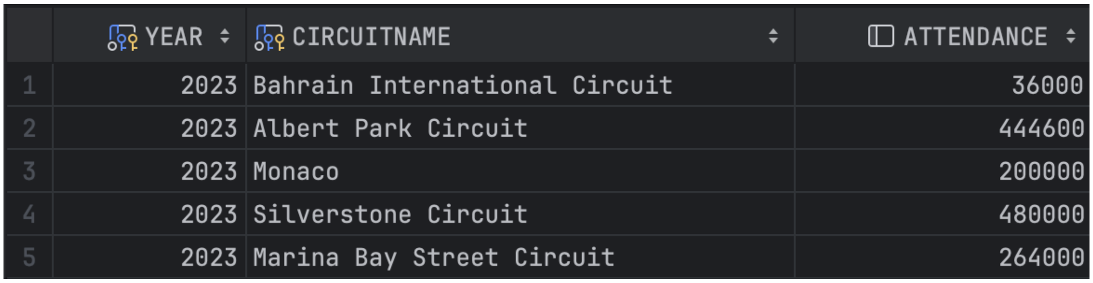
	CIRCUITNAME	CITY
1	Bahrain International Circuit	Sakhir
2	Albert Park Circuit	Melbourne
3	Monaco	Monte Carlo
4	Silverstone Circuit	Towcester
5	Marina Bay Street Circuit	Marina Bay

9. GrandPrix_3(**circuitName**, country)



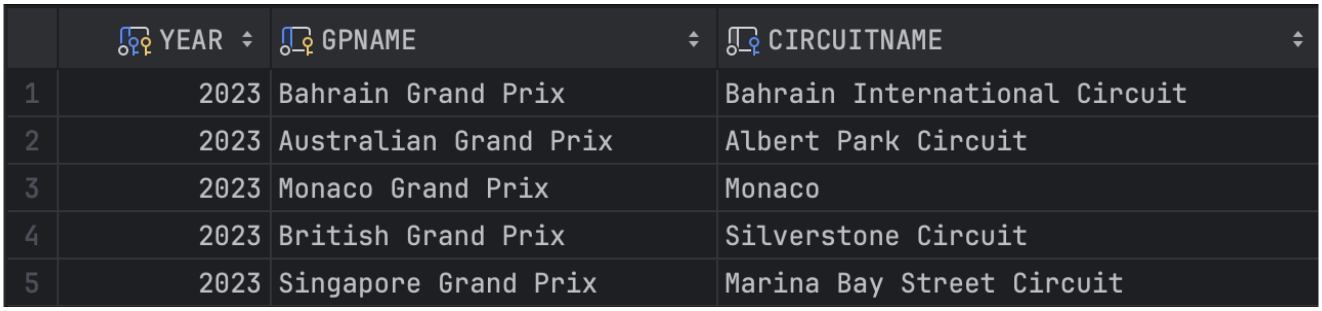
	CIRCUITNAME	COUNTRY
1	Bahrain International Circuit	Bahrain
2	Albert Park Circuit	Australia
3	Monaco	Monaco
4	Silverstone Circuit	England
5	Marina Bay Street Circuit	Singapore

10. GrandPrix_4(**year**, **circuitName**, attendance)



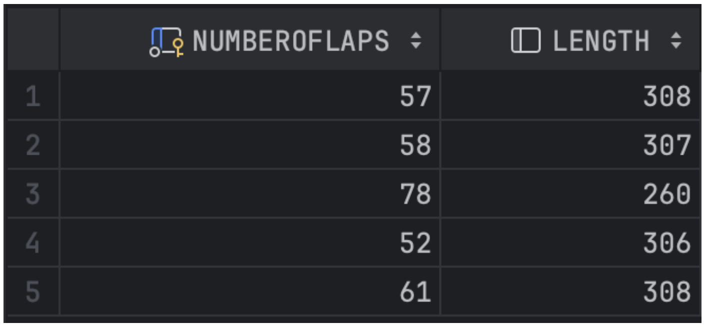
	YEAR	CIRCUITNAME	ATTENDANCE
1	2023	Bahrain International Circuit	36000
2	2023	Albert Park Circuit	444600
3	2023	Monaco	200000
4	2023	Silverstone Circuit	480000
5	2023	Marina Bay Street Circuit	264000

11. GrandPrix_5(**year**, gpName, **circuitName**)



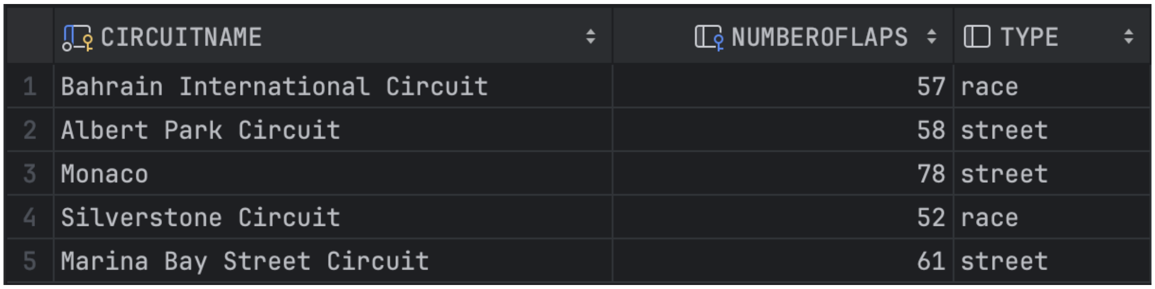
	YEAR	GPNAME	CIRCUITNAME
1	2023	Bahrain Grand Prix	Bahrain International Circuit
2	2023	Australian Grand Prix	Albert Park Circuit
3	2023	Monaco Grand Prix	Monaco
4	2023	British Grand Prix	Silverstone Circuit
5	2023	Singapore Grand Prix	Marina Bay Street Circuit

12. Circuit_Ref(numberOfLaps, length)



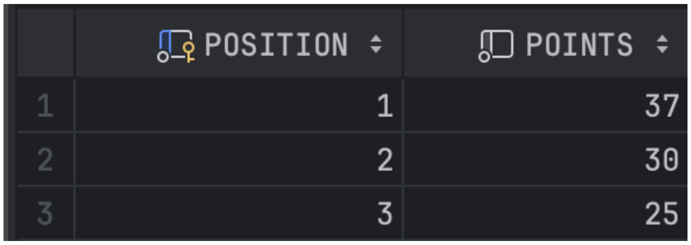
	NUMBEROFLAPS	LENGTH
1	57	308
2	58	307
3	78	260
4	52	306
5	61	308

13. Circuit_2(circuitName, **numberOfLaps,** type)



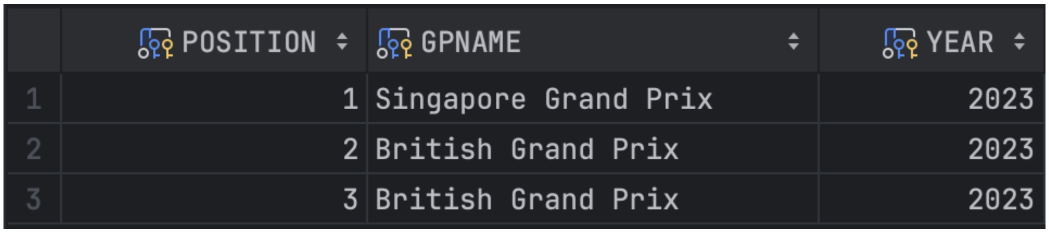
	CIRCUITNAME	NUMBEROFLAPS	TYPE
1	Bahrain International Circuit	57	race
2	Albert Park Circuit	58	street
3	Monaco	78	street
4	Silverstone Circuit	52	race
5	Marina Bay Street Circuit	61	street

14. GrandPrix_ConstructorStanding_Ref(position, points)



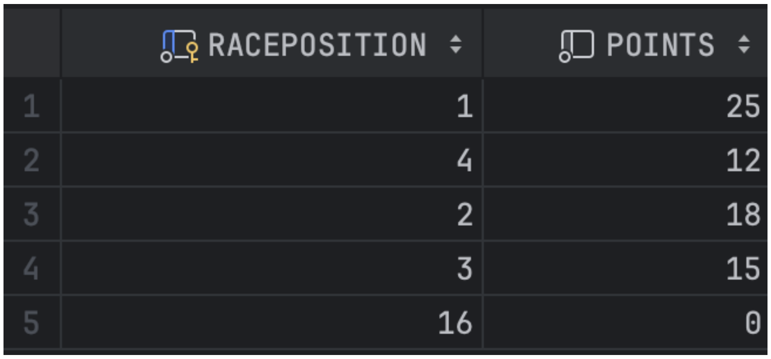
	POSITION	POINTS
1	1	37
2	2	30
3	3	25

15. GrandPrix_ConstructorStanding_2(**position**, **gpName**, **year**)



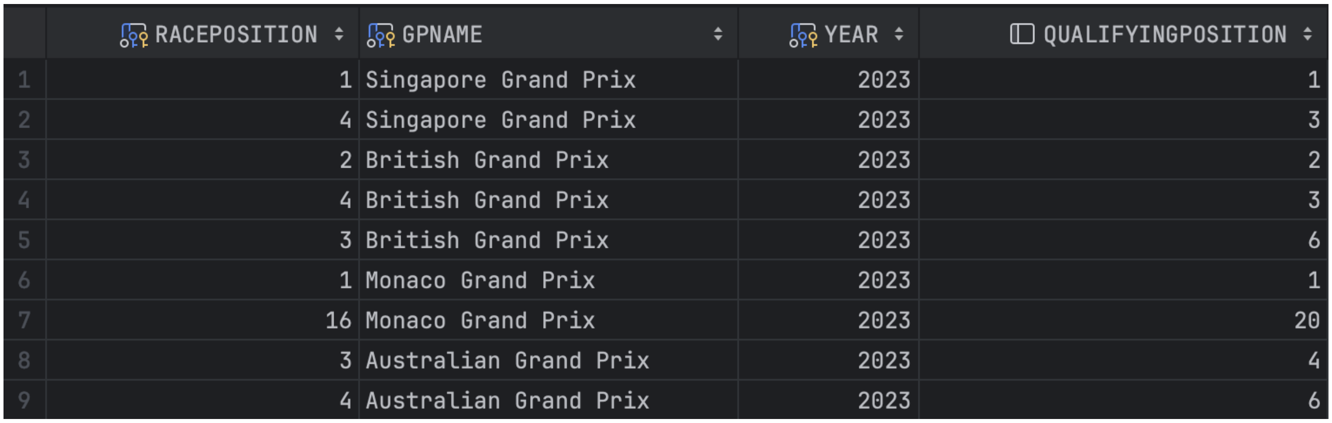
	POSITION	GPNAME	YEAR
1	1	Singapore Grand Prix	2023
2	2	British Grand Prix	2023
3	3	British Grand Prix	2023

16. GrandPrix_DriverStanding_Ref(racePosition, points)



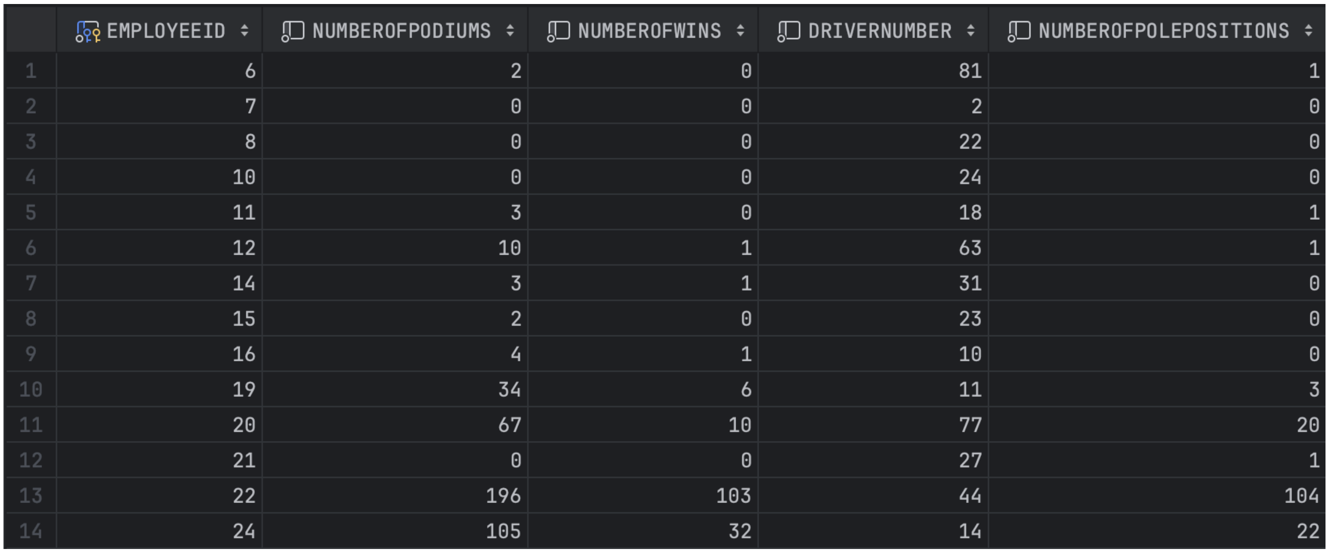
RACEPOSITION	POINTS
1	25
4	12
2	18
3	15
16	0

17. GrandPrix_DriverStanding_2(**racePosition**, **gpName**, **year**, qualifyingPosition)



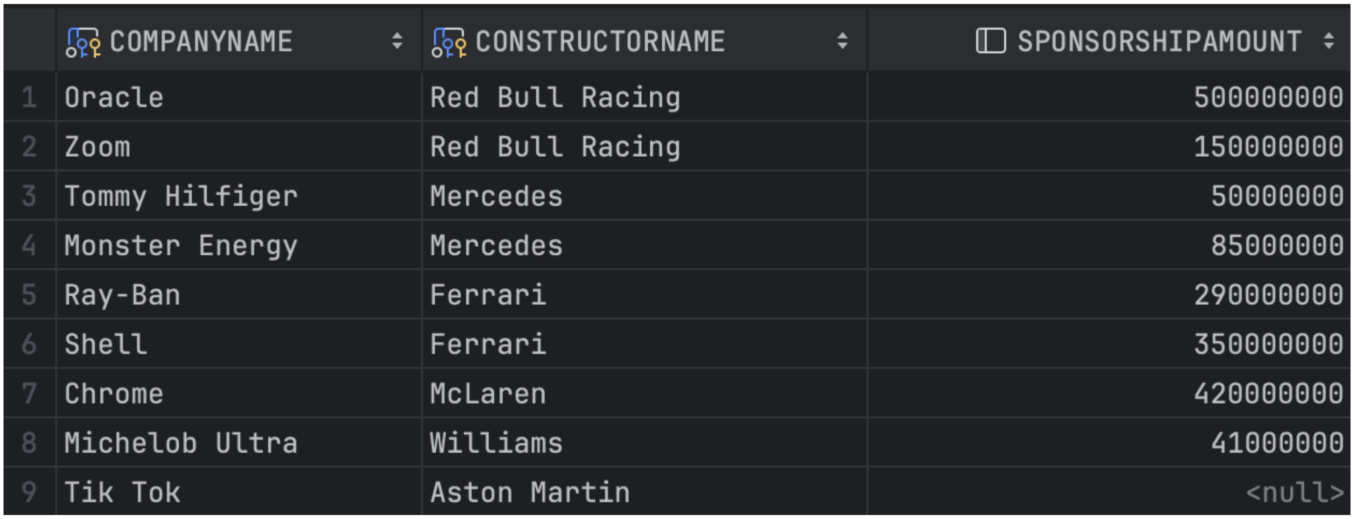
RACEPOSITION	GPNAME	YEAR	QUALIFYINGPOSITION
1	Singapore Grand Prix	2023	1
4	Singapore Grand Prix	2023	3
2	British Grand Prix	2023	2
4	British Grand Prix	2023	3
3	British Grand Prix	2023	6
1	Monaco Grand Prix	2023	1
16	Monaco Grand Prix	2023	20
3	Australian Grand Prix	2023	4
4	Australian Grand Prix	2023	6

18. Driver(**employeeId**, numberOfPodiums, numberOfWins, driverNumber, numberOfPolePositions)



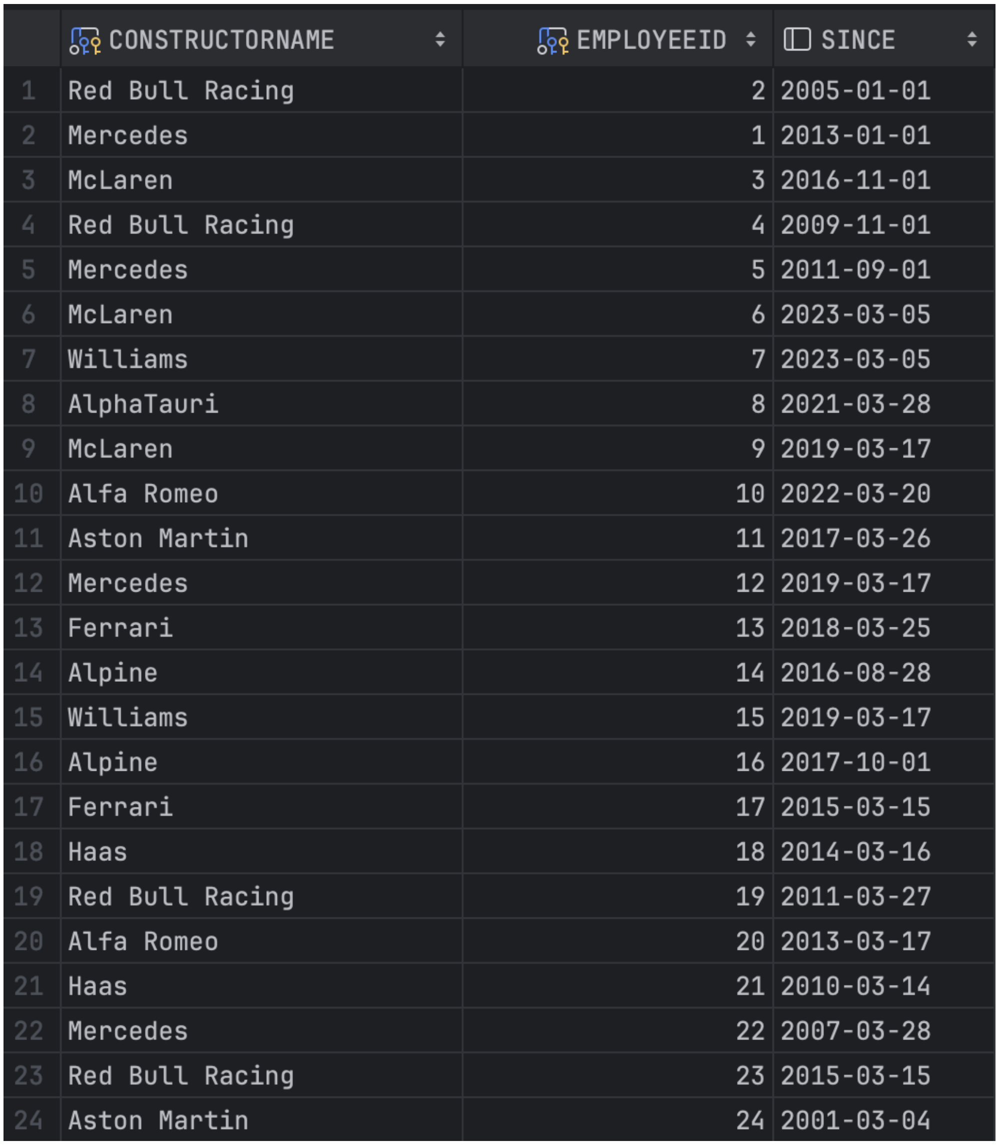
EMPLOYEEID	NUMBEROFPODIUMS	NUMBEROFWINS	DRIVERNUMBER	NUMBEROFPOLEPOSITIONS
6	2	0	81	1
7	0	0	2	0
8	0	0	22	0
10	0	0	24	0
11	3	0	18	1
12	10	1	63	1
14	3	1	31	0
15	2	0	23	0
16	4	1	10	0
19	34	6	11	3
20	67	10	77	20
21	0	0	27	1
22	196	103	44	104
24	105	32	14	22

19. Sponsors(**companyName**, **constructorName,** sponsorshipAmount)



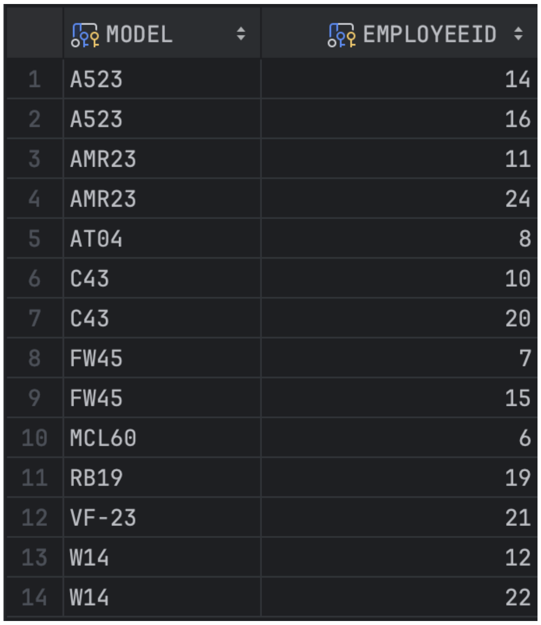
	COMPANYNAME	CONSTRUCTORNAME	SPONSORSHIPAMOUNT
1	Oracle	Red Bull Racing	500000000
2	Zoom	Red Bull Racing	150000000
3	Tommy Hilfiger	Mercedes	50000000
4	Monster Energy	Mercedes	85000000
5	Ray-Ban	Ferrari	290000000
6	Shell	Ferrari	350000000
7	Chrome	McLaren	420000000
8	Michelob Ultra	Williams	41000000
9	Tik Tok	Aston Martin	<null>

20. WorksWith(**constructorName**, **employeeId**, since)



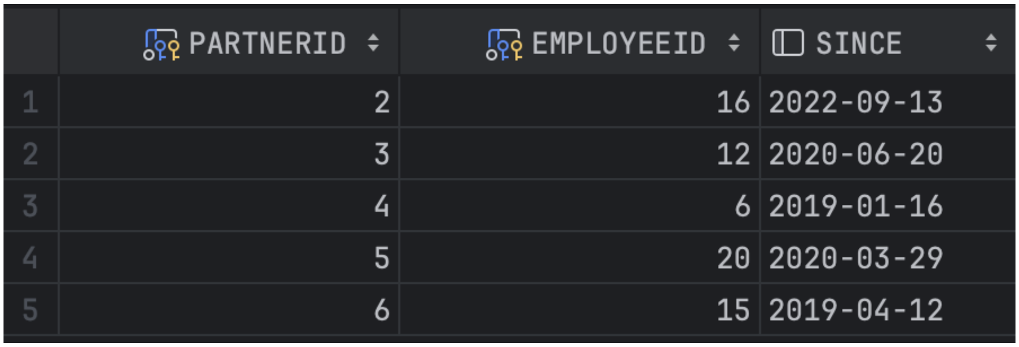
	CONSTRUCTORNAME	EMPLOYEEID	SINCE
1	Red Bull Racing	2	2005-01-01
2	Mercedes	1	2013-01-01
3	McLaren	3	2016-11-01
4	Red Bull Racing	4	2009-11-01
5	Mercedes	5	2011-09-01
6	McLaren	6	2023-03-05
7	Williams	7	2023-03-05
8	AlphaTauri	8	2021-03-28
9	McLaren	9	2019-03-17
10	Alfa Romeo	10	2022-03-20
11	Aston Martin	11	2017-03-26
12	Mercedes	12	2019-03-17
13	Ferrari	13	2018-03-25
14	Alpine	14	2016-08-28
15	Williams	15	2019-03-17
16	Alpine	16	2017-10-01
17	Ferrari	17	2015-03-15
18	Haas	18	2014-03-16
19	Red Bull Racing	19	2011-03-27
20	Alfa Romeo	20	2013-03-17
21	Haas	21	2010-03-14
22	Mercedes	22	2007-03-28
23	Red Bull Racing	23	2015-03-15
24	Aston Martin	24	2001-03-04

21. Drives(**model**, **employeeId**)



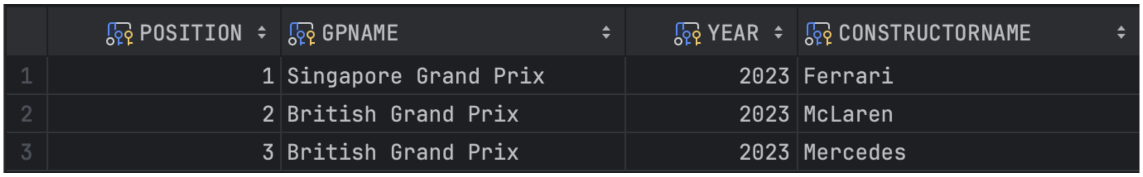
	MODEL	EMPLOYEEID
1	A523	14
2	A523	16
3	AMR23	11
4	AMR23	24
5	AT04	8
6	C43	10
7	C43	20
8	FW45	7
9	FW45	15
10	MCL60	6
11	RB19	19
12	VF-23	21
13	W14	12
14	W14	22

22. InRelationshipWith(**partnerId**, **employeeId**, since**)**



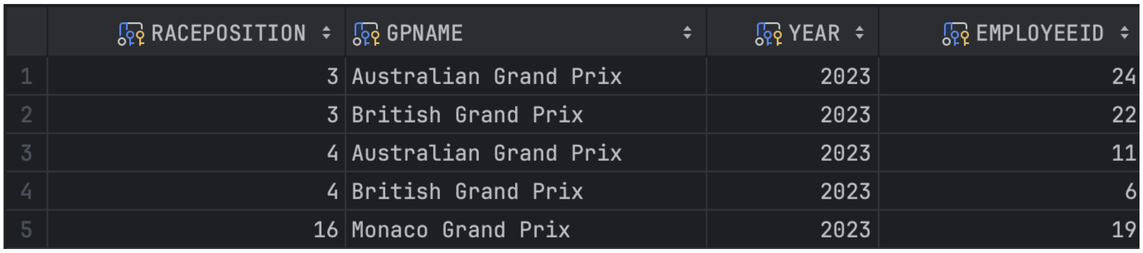
	PARTNERID	EMPLOYEEID	SINCE
1	2	16	2022-09-13
2	3	12	2020-06-20
3	4	6	2019-01-16
4	5	20	2020-03-29
5	6	15	2019-04-12

23. ConstructorHolds(**position**, **gpName**, **year**, **constructorName**)



	POSITION	GPNAME	YEAR	CONSTRUCTORNAME
1	1	Singapore Grand Prix	2023	Ferrari
2	2	British Grand Prix	2023	McLaren
3	3	British Grand Prix	2023	Mercedes

24. DriverHolds(**racePosition**, **gpName**, **year**, **employeeId**)



	RACEPOSITION	GPNAME	YEAR	EMPLOYEEID
1	3	Australian Grand Prix	2023	24
2	3	British Grand Prix	2023	22
3	4	Australian Grand Prix	2023	11
4	4	British Grand Prix	2023	6
5	16	Monaco Grand Prix	2023	19

SQL Queries:

Insert:

```
driver.php, lines 78-82
```

Delete:

`driver.php, line 144`

Update:

`constructor.php, line 46`

Selection:

`grandprix.php, lines 49-88`

Projection:

`home.php, lines 46-91`

Join:

`driver.php, lines 112-128`

Aggregation with Group By:

`constructor.php, lines 58-61`

Aggregation with Having:

`constructor.php, lines 70-80`

Nested Aggregation with Group By:

`grandprix.php, lines 105-113`

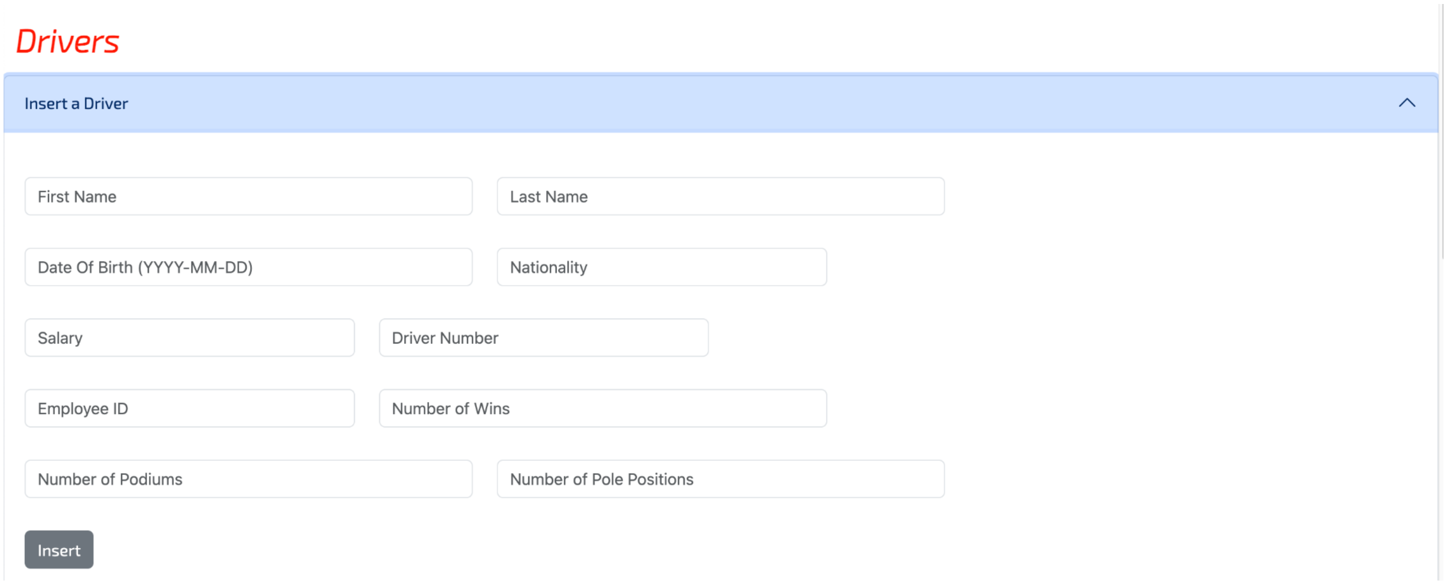
Division:

`constructor.php, lines 91-106`

Query Functionalities:

Insert:

Before:



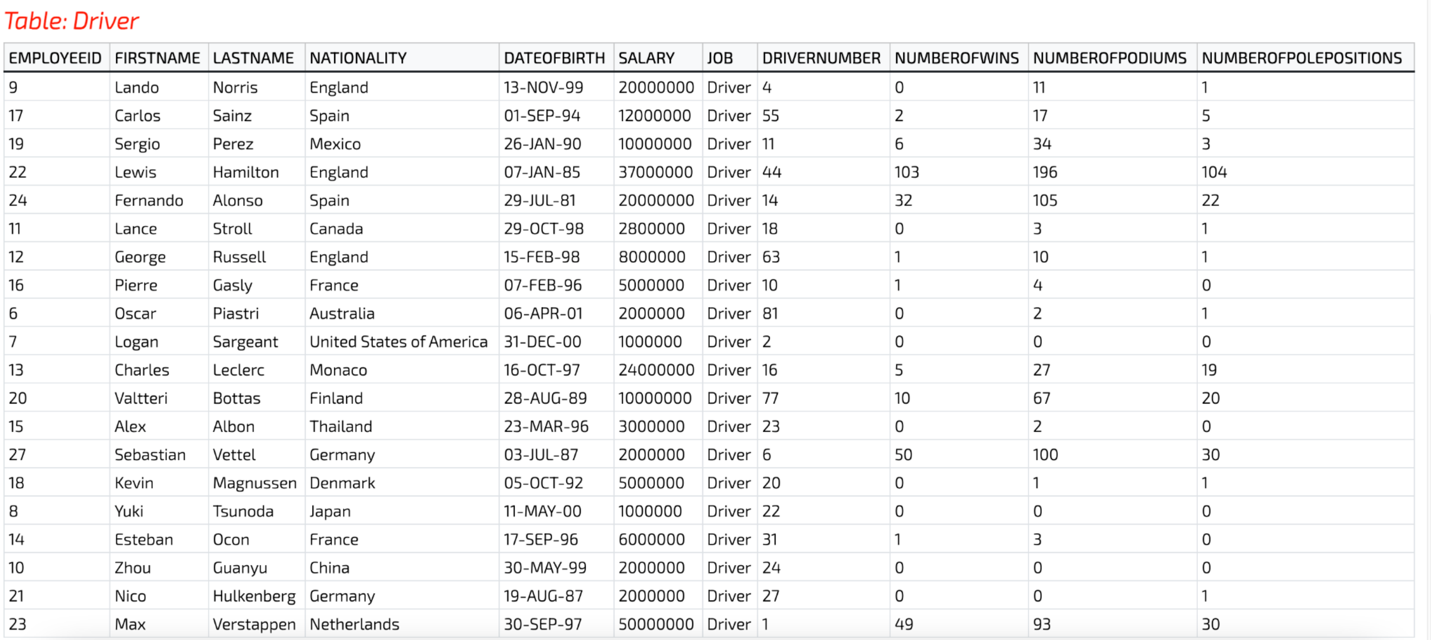
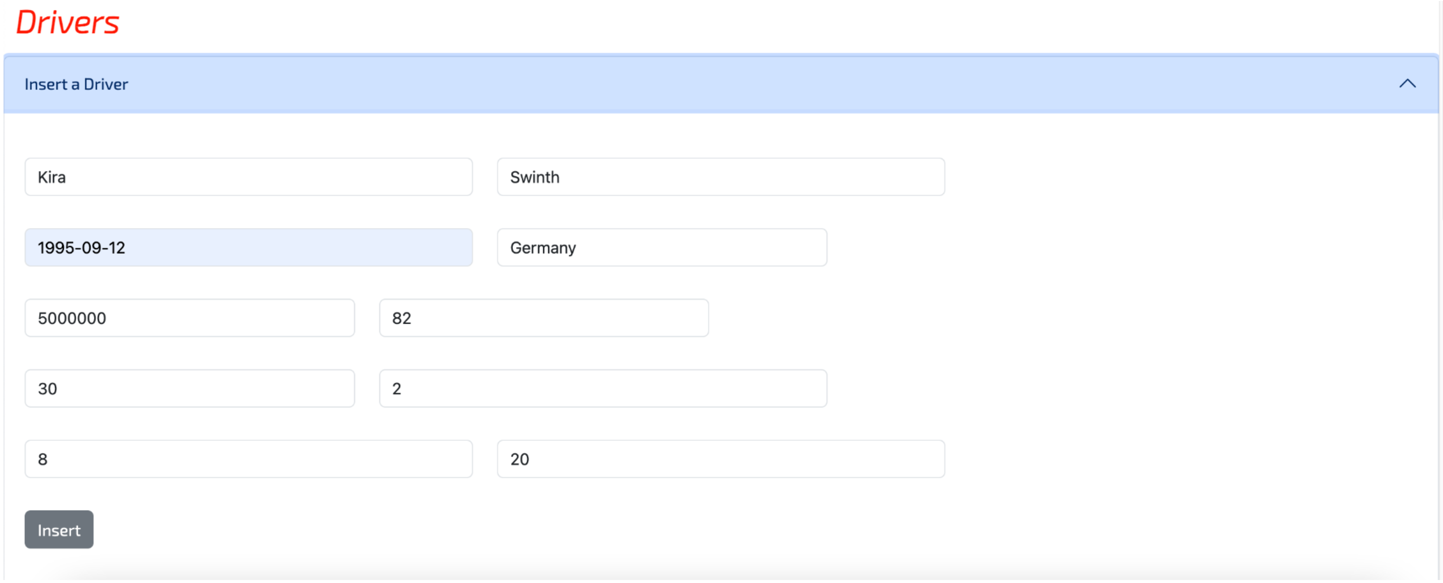


Table: Driver

EMPLOYEEID	FIRSTNAME	LASTNAME	NATIONALITY	DATEOFBIRTH	SALARY	JOB	DRIVERNUMBER	NUMBEROFWINS	NUMBEROFPODIUMS	NUMBEROFPOLEPOSITIONS
9	Lando	Norris	England	13-NOV-99	20000000	Driver	4	0	11	1
17	Carlos	Sainz	Spain	01-SEP-94	12000000	Driver	55	2	17	5
19	Sergio	Perez	Mexico	26-JAN-90	10000000	Driver	11	6	34	3
22	Lewis	Hamilton	England	07-JAN-85	37000000	Driver	44	103	196	104
24	Fernando	Alonso	Spain	29-JUL-81	20000000	Driver	14	32	105	22
11	Lance	Stroll	Canada	29-OCT-98	2800000	Driver	18	0	3	1
12	George	Russell	England	15-FEB-98	8000000	Driver	63	1	10	1
16	Pierre	Gasly	France	07-FEB-96	5000000	Driver	10	1	4	0
6	Oscar	Piastri	Australia	06-APR-01	2000000	Driver	81	0	2	1
7	Logan	Sargeant	United States of America	31-DEC-00	1000000	Driver	2	0	0	0
13	Charles	Leclerc	Monaco	16-OCT-97	24000000	Driver	16	5	27	19
20	Valtteri	Bottas	Finland	28-AUG-89	10000000	Driver	77	10	67	20
15	Alex	Albon	Thailand	23-MAR-96	3000000	Driver	23	0	2	0
27	Sebastian	Vettel	Germany	03-JUL-87	2000000	Driver	6	50	100	30
18	Kevin	Magnussen	Denmark	05-OCT-92	5000000	Driver	20	0	1	1
8	Yuki	Tsunoda	Japan	11-MAY-00	1000000	Driver	22	0	0	0
14	Esteban	Ocon	France	17-SEP-96	6000000	Driver	31	1	3	0
10	Zhou	Guanyu	China	30-MAY-99	2000000	Driver	24	0	0	0
21	Nico	Hulkenberg	Germany	19-AUG-87	2000000	Driver	27	0	0	1
23	Max	Verstappen	Netherlands	30-SEP-97	50000000	Driver	1	49	93	30

During:



Drivers

Kira

Swinth

1995-09-12

Germany

5000000

82

30

2

8

20

After:

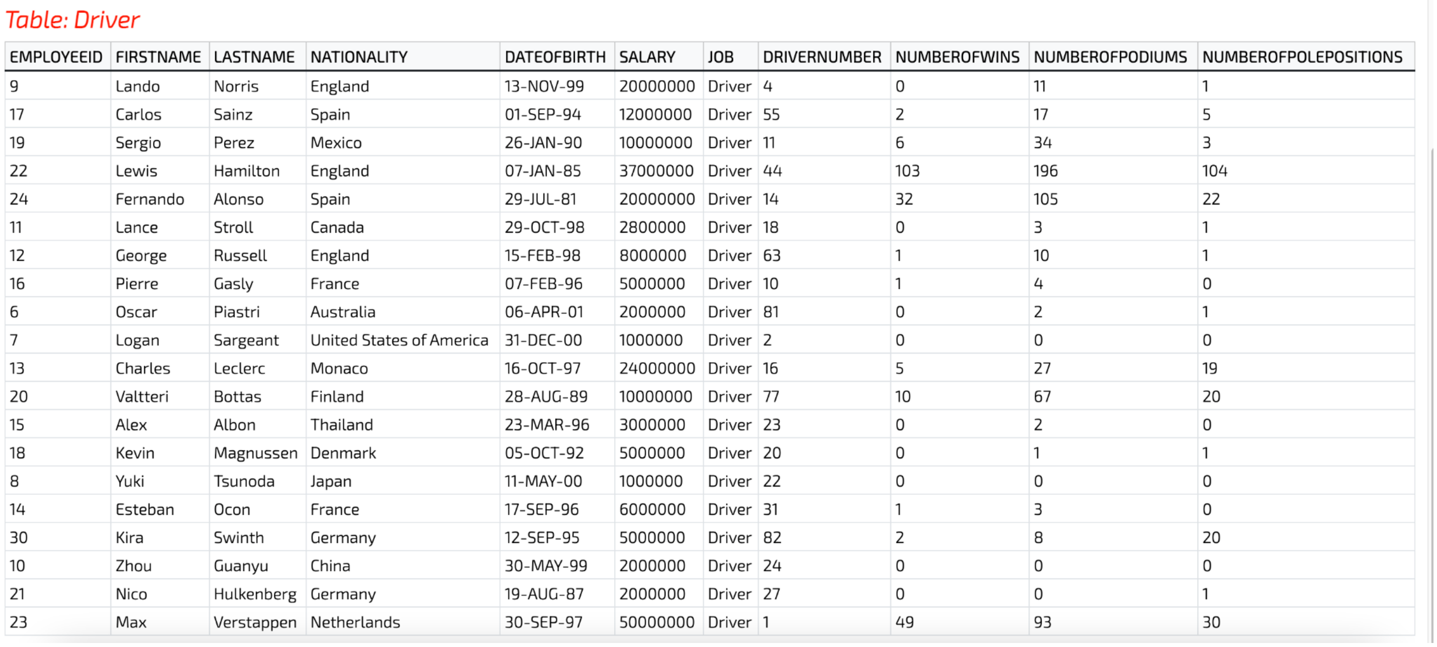
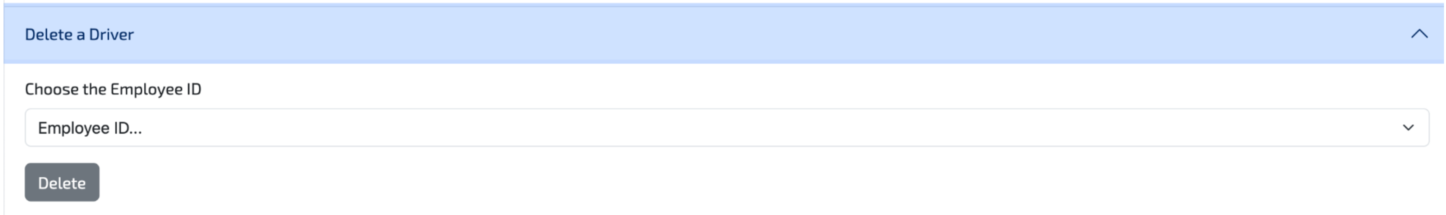


Table: Driver

EMPLOYEEID	FIRSTNAME	LASTNAME	NATIONALITY	DATEOFBIRTH	SALARY	JOB	DRIVERNUMBER	NUMBEROFWINS	NUMBEROFPODIUMS	NUMBEROFPOLEPOSITIONS
9	Lando	Norris	England	13-NOV-99	20000000	Driver	4	0	11	1
17	Carlos	Sainz	Spain	01-SEP-94	12000000	Driver	55	2	17	5
19	Sergio	Perez	Mexico	26-JAN-90	10000000	Driver	11	6	34	3
22	Lewis	Hamilton	England	07-JAN-85	37000000	Driver	44	103	196	104
24	Fernando	Alonso	Spain	29-JUL-81	20000000	Driver	14	32	105	22
11	Lance	Stroll	Canada	29-OCT-98	2800000	Driver	18	0	3	1
12	George	Russell	England	15-FEB-98	8000000	Driver	63	1	10	1
16	Pierre	Gasly	France	07-FEB-96	5000000	Driver	10	1	4	0
6	Oscar	Piastri	Australia	06-APR-01	2000000	Driver	81	0	2	1
7	Logan	Sargeant	United States of America	31-DEC-00	1000000	Driver	2	0	0	0
13	Charles	Leclerc	Monaco	16-OCT-97	24000000	Driver	16	5	27	19
20	Valtteri	Bottas	Finland	28-AUG-89	10000000	Driver	77	10	67	20
15	Alex	Albon	Thailand	23-MAR-96	3000000	Driver	23	0	2	0
18	Kevin	Magnussen	Denmark	05-OCT-92	5000000	Driver	20	0	1	1
8	Yuki	Tsunoda	Japan	11-MAY-00	1000000	Driver	22	0	0	0
14	Esteban	Ocon	France	17-SEP-96	6000000	Driver	31	1	3	0
30	Kira	Swinth	Germany	12-SEP-95	5000000	Driver	82	2	8	20
10	Zhou	Guanyu	China	30-MAY-99	2000000	Driver	24	0	0	0
21	Nico	Hulkenberg	Germany	19-AUG-87	2000000	Driver	27	0	0	1
23	Max	Verstappen	Netherlands	30-SEP-97	50000000	Driver	1	49	93	30

Delete:

Before:



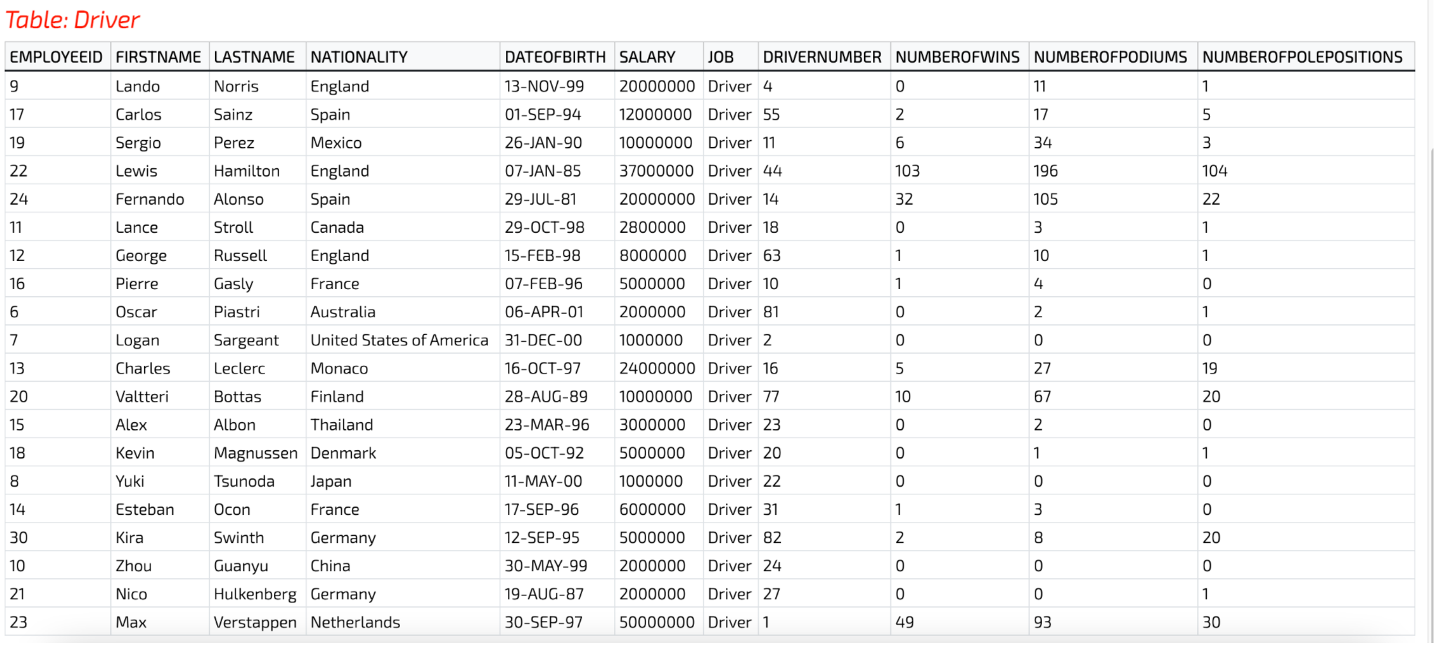
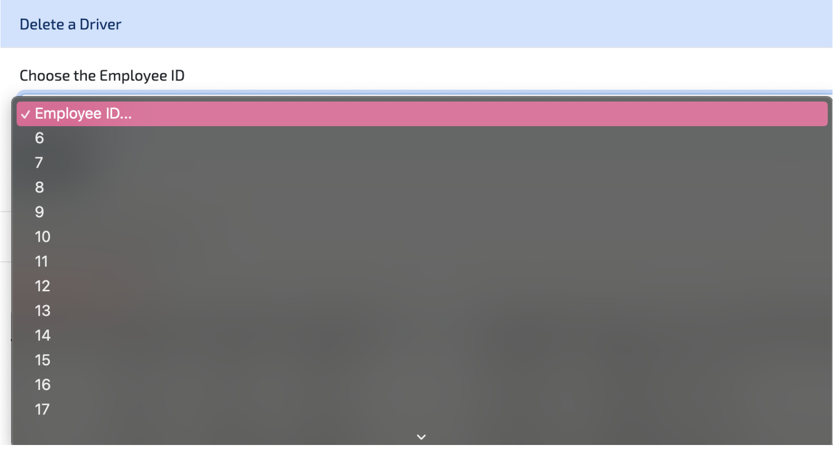


Table: Driver

EMPLOYEEID	FIRSTNAME	LASTNAME	NATIONALITY	DATEOFBIRTH	SALARY	JOB	DRIVERNUMBER	NUMBEROFWINS	NUMBEROFPODIUMS	NUMBEROFPOLEPOSITIONS
9	Lando	Norris	England	13-NOV-99	20000000	Driver	4	0	11	1
17	Carlos	Sainz	Spain	01-SEP-94	12000000	Driver	55	2	17	5
19	Sergio	Perez	Mexico	26-JAN-90	10000000	Driver	11	6	34	3
22	Lewis	Hamilton	England	07-JAN-85	37000000	Driver	44	103	196	104
24	Fernando	Alonso	Spain	29-JUL-81	20000000	Driver	14	32	105	22
11	Lance	Stroll	Canada	29-OCT-98	2800000	Driver	18	0	3	1
12	George	Russell	England	15-FEB-98	8000000	Driver	63	1	10	1
16	Pierre	Gasly	France	07-FEB-96	5000000	Driver	10	1	4	0
6	Oscar	Piastri	Australia	06-APR-01	2000000	Driver	81	0	2	1
7	Logan	Sargeant	United States of America	31-DEC-00	1000000	Driver	2	0	0	0
13	Charles	Leclerc	Monaco	16-OCT-97	24000000	Driver	16	5	27	19
20	Valtteri	Bottas	Finland	28-AUG-89	10000000	Driver	77	10	67	20
15	Alex	Albon	Thailand	23-MAR-96	3000000	Driver	23	0	2	0
18	Kevin	Magnussen	Denmark	05-OCT-92	5000000	Driver	20	0	1	1
8	Yuki	Tsunoda	Japan	11-MAY-00	1000000	Driver	22	0	0	0
14	Esteban	Ocon	France	17-SEP-96	6000000	Driver	31	1	3	0
30	Kira	Swinth	Germany	12-SEP-95	5000000	Driver	82	2	8	20
10	Zhou	Guanyu	China	30-MAY-99	2000000	Driver	24	0	0	0
21	Nico	Hulkenberg	Germany	19-AUG-87	2000000	Driver	27	0	0	1
23	Max	Verstappen	Netherlands	30-SEP-97	50000000	Driver	1	49	93	30

During: Delete EmployeeID = 6



Delete a Driver

Choose the Employee ID

✓ Employee ID...
6
7
8
9
10
11
12
13
14
15
16
17

After:

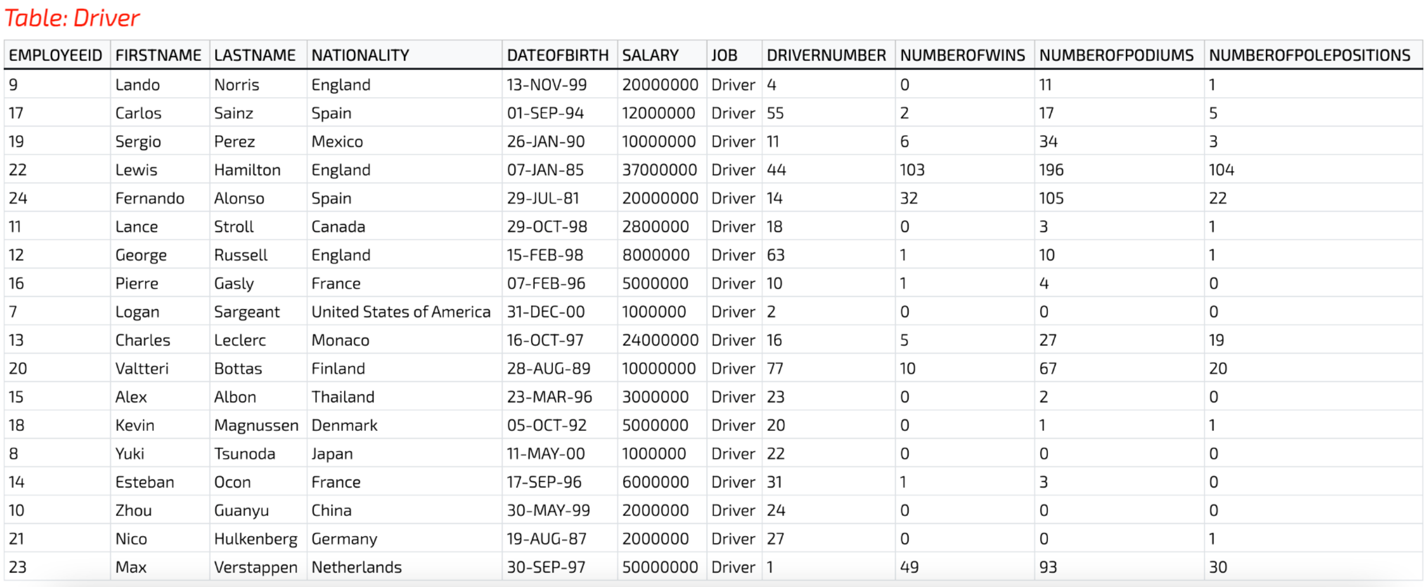
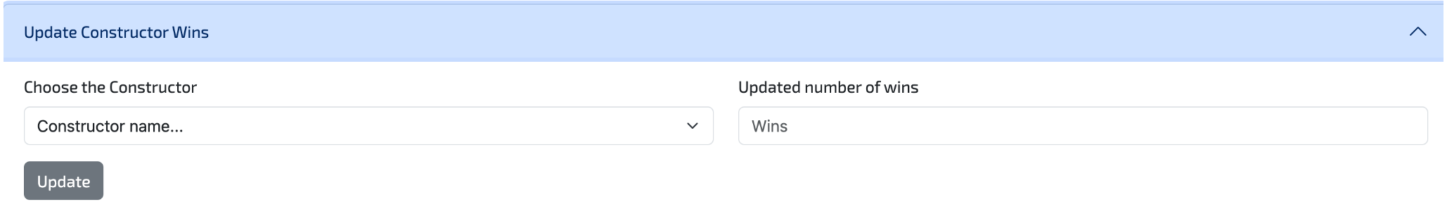


Table: Driver

EMPLOYEEID	FIRSTNAME	LASTNAME	NATIONALITY	DATEOFBIRTH	SALARY	JOB	DRIVERNUMBER	NUMBEROFWINS	NUMBEROFPODIUMS	NUMBEROFPOLEPOSITIONS
9	Lando	Norris	England	13-NOV-99	20000000	Driver	4	0	11	1
17	Carlos	Sainz	Spain	01-SEP-94	12000000	Driver	55	2	17	5
19	Sergio	Perez	Mexico	26-JAN-90	10000000	Driver	11	6	34	3
22	Lewis	Hamilton	England	07-JAN-85	37000000	Driver	44	103	196	104
24	Fernando	Alonso	Spain	29-JUL-81	20000000	Driver	14	32	105	22
11	Lance	Stroll	Canada	29-OCT-98	2800000	Driver	18	0	3	1
12	George	Russell	England	15-FEB-98	8000000	Driver	63	1	10	1
16	Pierre	Gasly	France	07-FEB-96	5000000	Driver	10	1	4	0
7	Logan	Sargeant	United States of America	31-DEC-00	1000000	Driver	2	0	0	0
13	Charles	Leclerc	Monaco	16-OCT-97	24000000	Driver	16	5	27	19
20	Valtteri	Bottas	Finland	28-AUG-89	10000000	Driver	77	10	67	20
15	Alex	Albon	Thailand	23-MAR-96	3000000	Driver	23	0	2	0
18	Kevin	Magnussen	Denmark	05-OCT-92	5000000	Driver	20	0	1	1
8	Yuki	Tsunoda	Japan	11-MAY-00	1000000	Driver	22	0	0	0
14	Esteban	Ocon	France	17-SEP-96	6000000	Driver	31	1	3	0
10	Zhou	Guanyu	China	30-MAY-99	2000000	Driver	24	0	0	0
21	Nico	Hulkenberg	Germany	19-AUG-87	2000000	Driver	27	0	0	1
23	Max	Verstappen	Netherlands	30-SEP-97	50000000	Driver	1	49	93	30

Update:

Before:



Update Constructor Wins

Choose the Constructor: Constructor name...

Updated number of wins: Wins

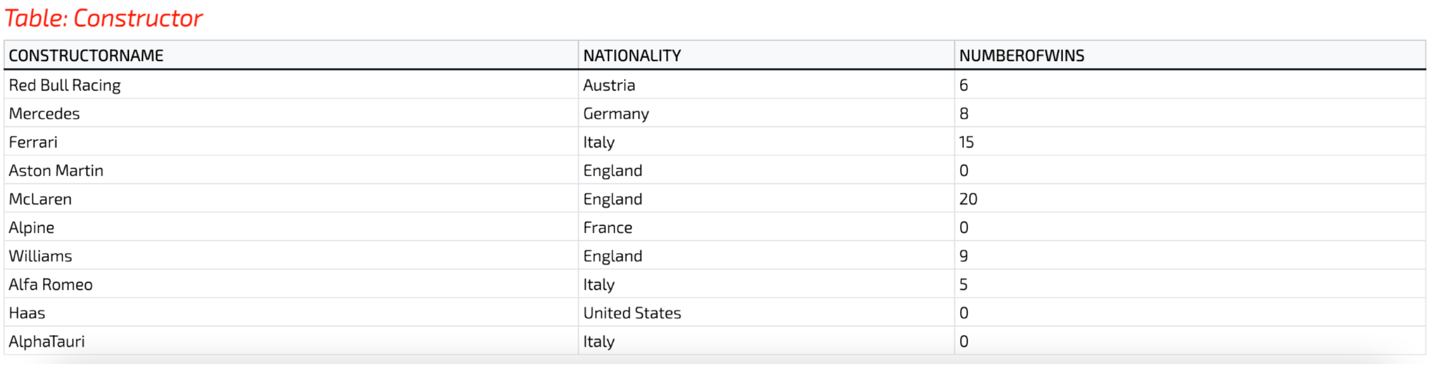
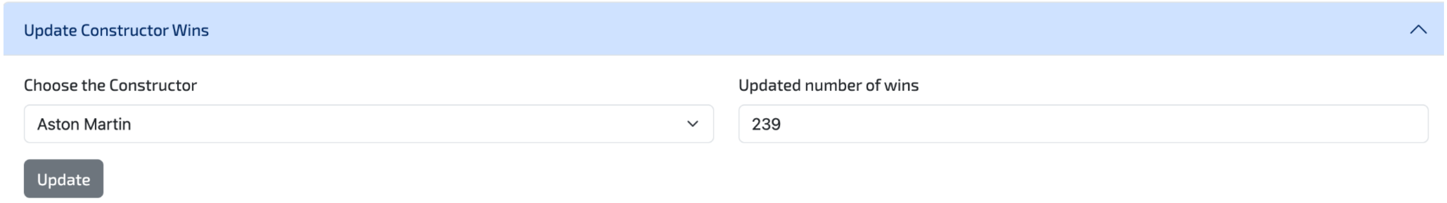


Table: Constructor

CONSTRUCTORNAME	NATIONALITY	NUMBEROFWINS
Red Bull Racing	Austria	6
Mercedes	Germany	8
Ferrari	Italy	15
Aston Martin	England	0
McLaren	England	20
Alpine	France	0
Williams	England	9
Alfa Romeo	Italy	5
Haas	United States	0
AlphaTauri	Italy	0

During:



Update Constructor Wins

Choose the Constructor: Aston Martin

Updated number of wins: 239

After:

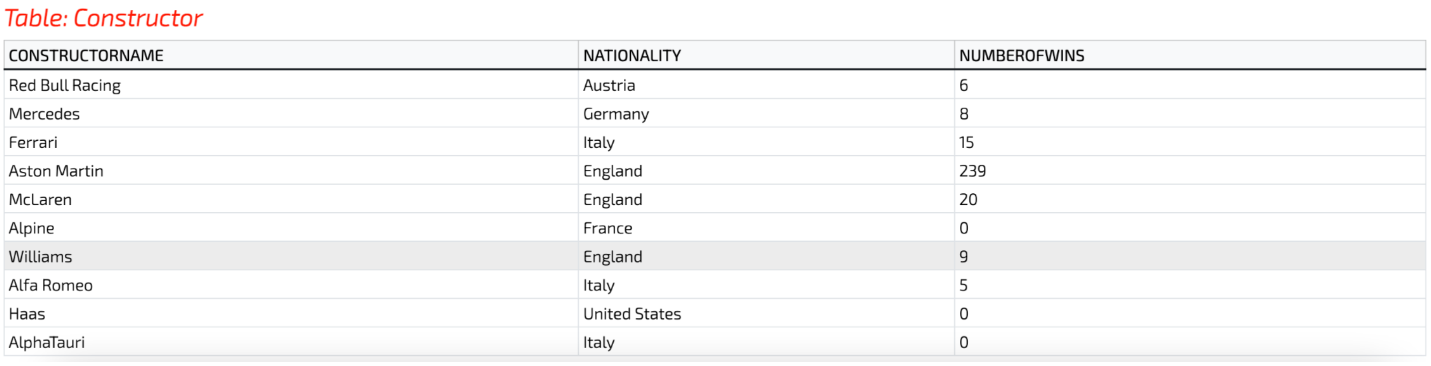
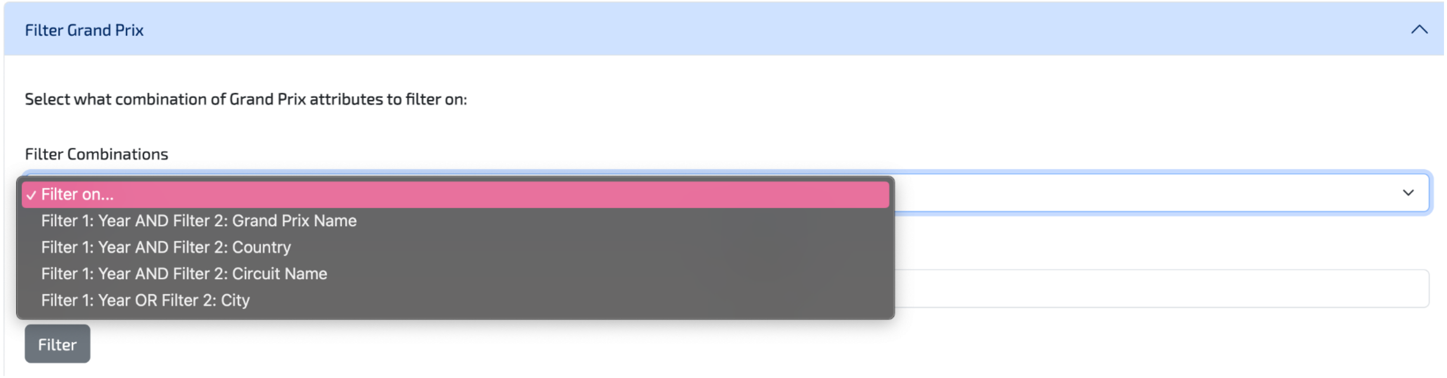


Table: Constructor

CONSTRUCTORNAME	NATIONALITY	NUMBEROFWINS
Red Bull Racing	Austria	6
Mercedes	Germany	8
Ferrari	Italy	15
Aston Martin	England	239
McLaren	England	20
Alpine	France	0
Williams	England	9
Alfa Romeo	Italy	5
Haas	United States	0
AlphaTauri	Italy	0

Selection:

Before:



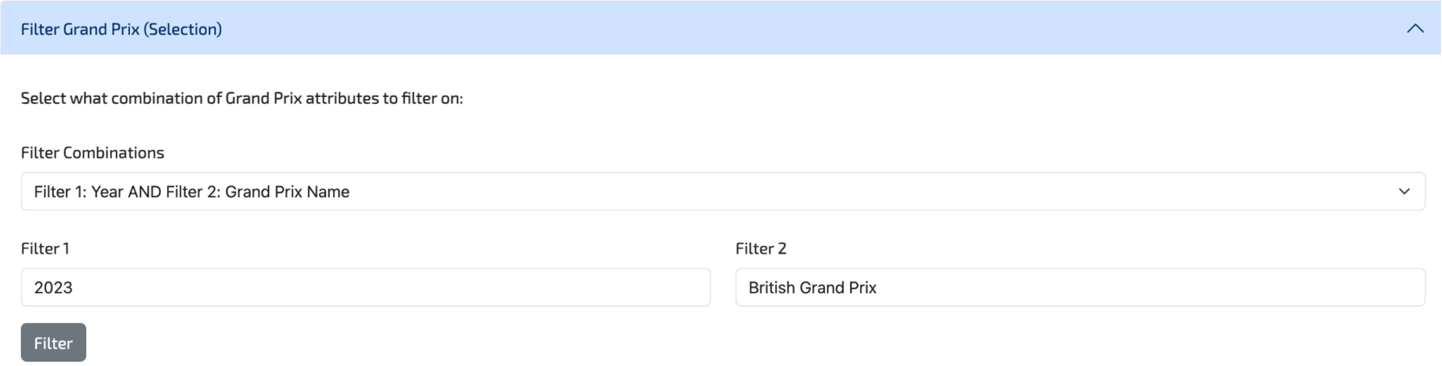
Filter Grand Prix

Select what combination of Grand Prix attributes to filter on:

Filter Combinations

✓ Filter on...
Filter 1: Year AND Filter 2: Grand Prix Name
Filter 1: Year AND Filter 2: Country
Filter 1: Year AND Filter 2: Circuit Name
Filter 1: Year OR Filter 2: City
Filter

During:



Filter Grand Prix (Selection)

Select what combination of Grand Prix attributes to filter on:

Filter Combinations: Filter 1: Year AND Filter 2: Grand Prix Name

Filter 1: 2023

Filter 2: British Grand Prix

After:

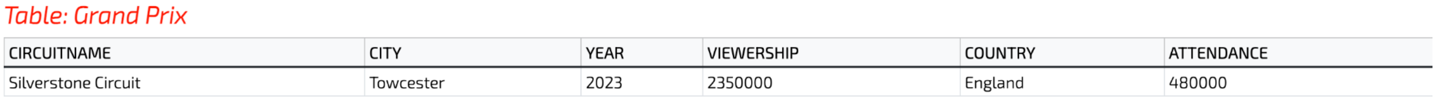
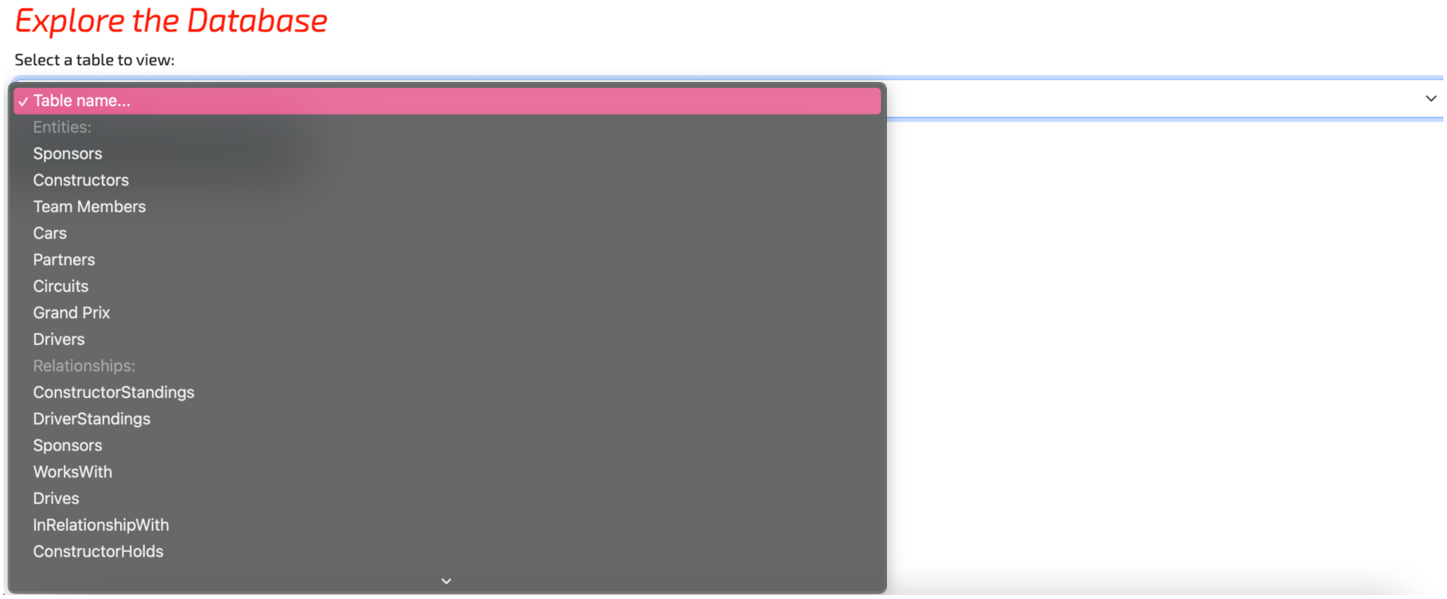


Table: Grand Prix

CIRCUITNAME	CITY	YEAR	VIEWERSHIP	COUNTRY	ATTENDANCE
Silverstone Circuit	Towcester	2023	2350000	England	480000

Projection:

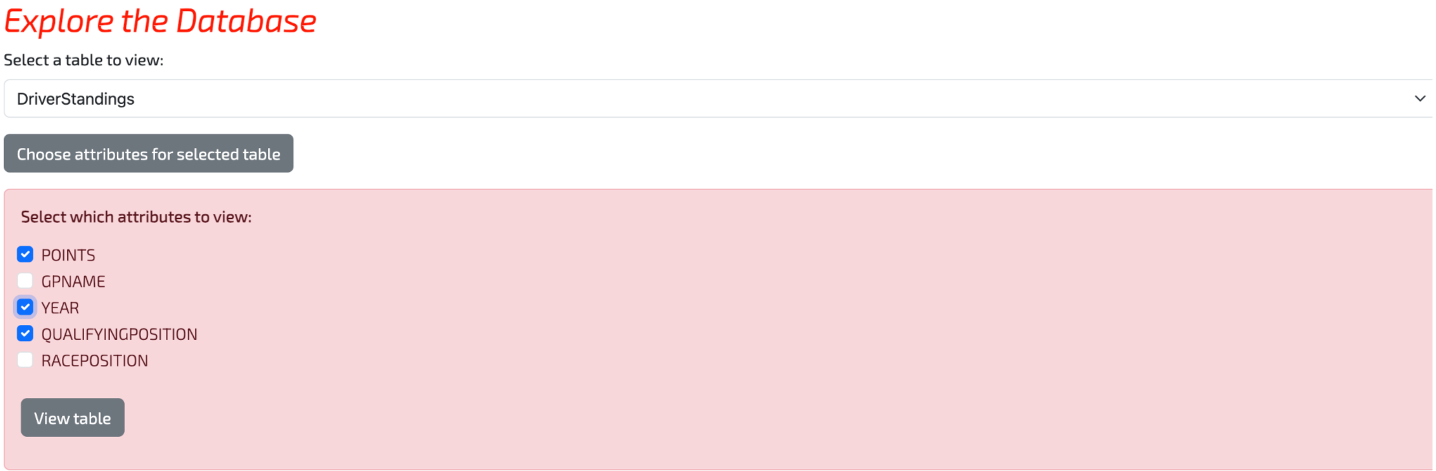
Before:



Explore the Database

Select a table to view:

During:



Explore the Database

Select a table to view: DriverStandings

Select which attributes to view:

☑ POINTS
☐ GPNAME
☑ YEAR
☑ QUALIFYINGPOSITION
☐ RACEPOSITION

After:

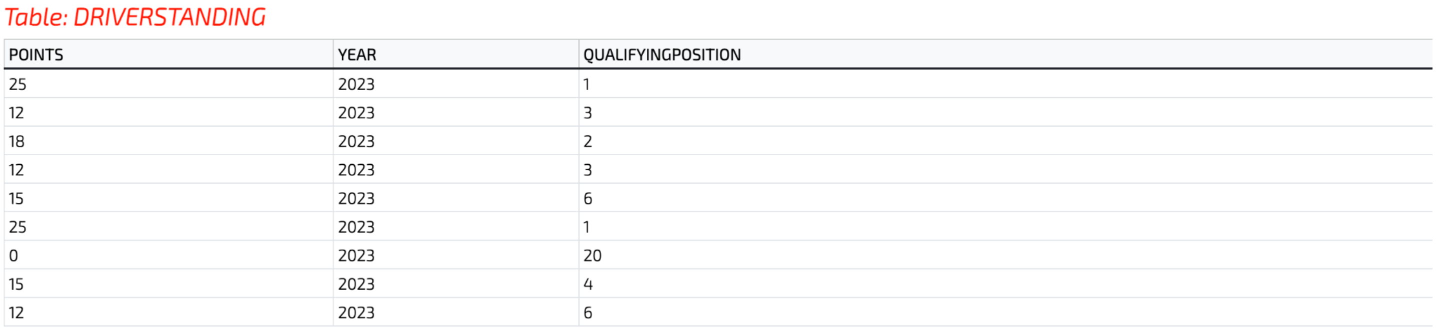
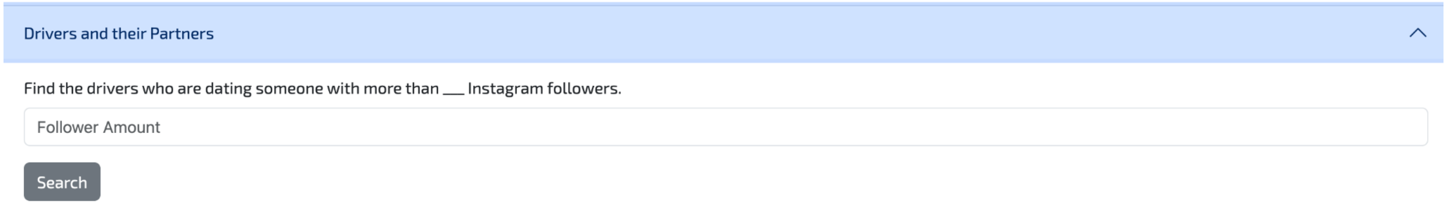


Table: DRIVERSTANDING

POINTS	YEAR	QUALIFYINGPOSITION
25	2023	1
12	2023	3
18	2023	2
12	2023	3
15	2023	6
25	2023	1
0	2023	20
15	2023	4
12	2023	6

Join:

Before:



Find the drivers who are dating someone with more than ___ Instagram followers.

After:

With Follower Amount = 1000:

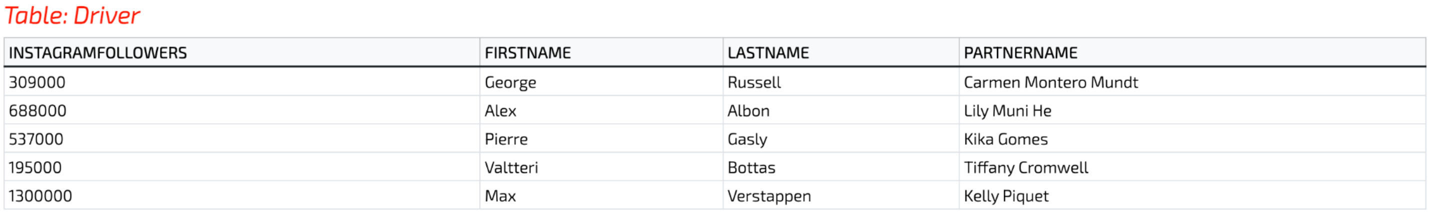
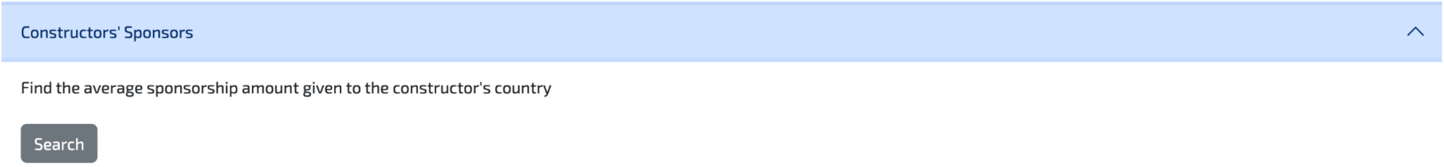


Table: Driver

INSTAGRAMFOLLOWERS	FIRSTNAME	LASTNAME	PARTNERNAME
309000	George	Russell	Carmen Montero Mundt
688000	Alex	Albon	Lily Muni He
537000	Pierre	Gasly	Kika Gomes
195000	Valtteri	Bottas	Tiffany Cromwell
1300000	Max	Verstappen	Kelly Piquet

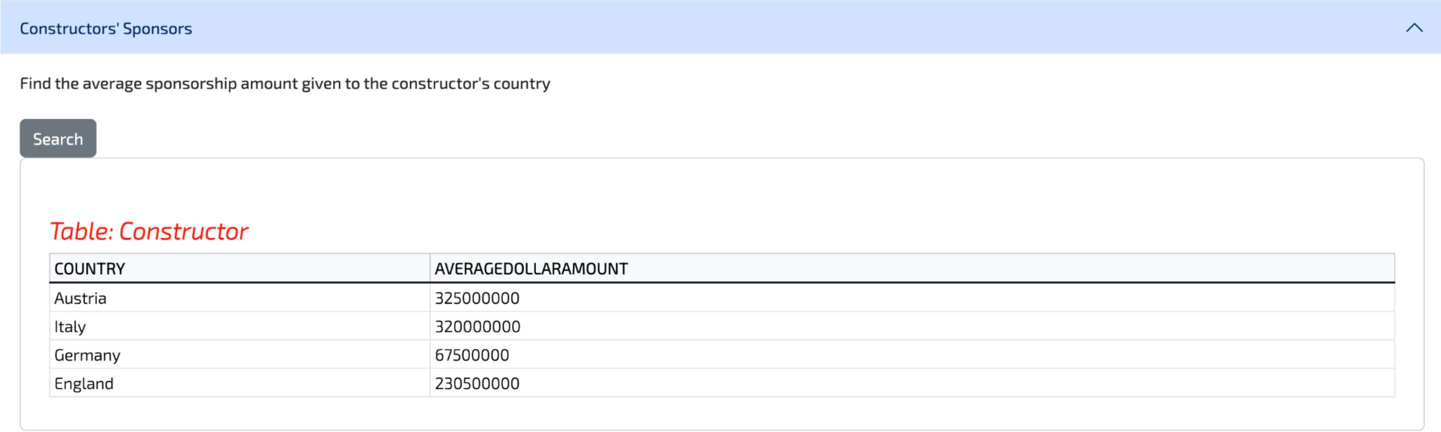
Aggregation with Group By:

Before:



Find the average sponsorship amount given to the constructor's country

After**:**



Constructors' Sponsors

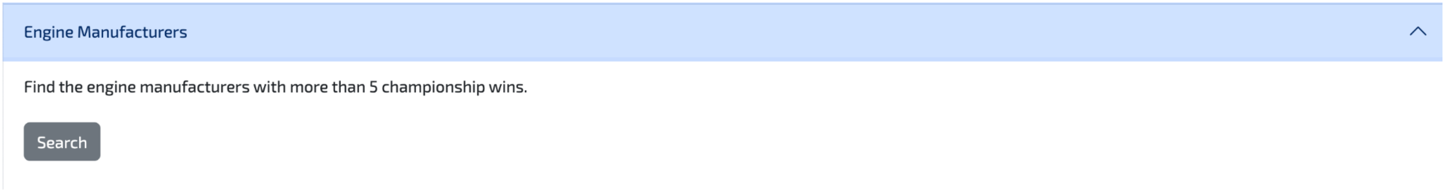
Find the average sponsorship amount given to the constructor's country

Table: Constructor

COUNTRY	AVERAGEDOLLARAMOUNT
Austria	325000000
Italy	320000000
Germany	67500000
England	230500000

Aggregation with Having:

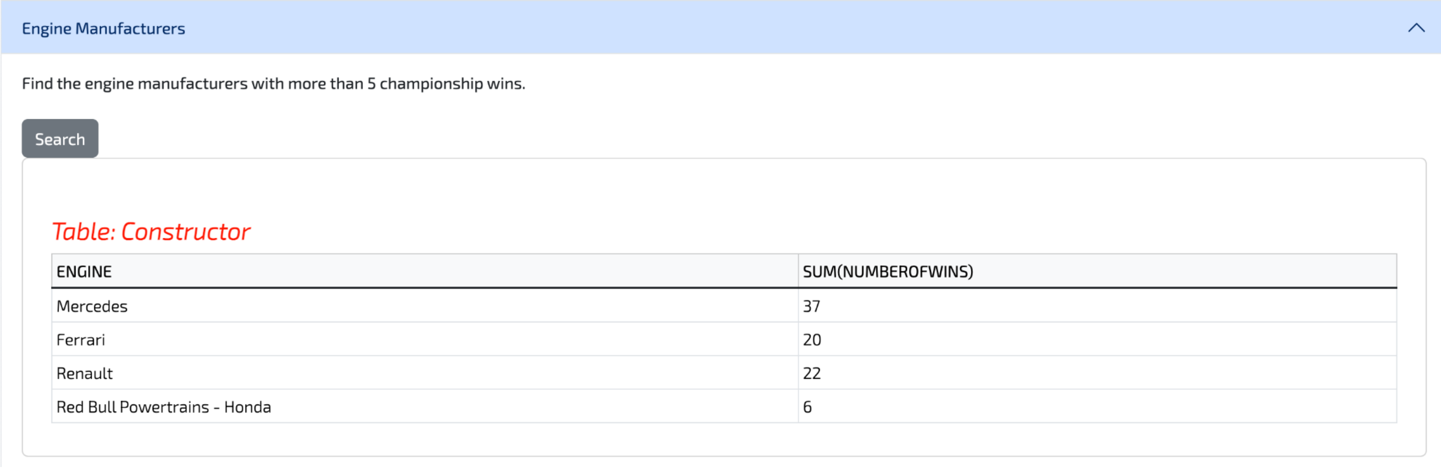
Before:



Engine Manufacturers

Find the engine manufacturers with more than 5 championship wins.

After:



Engine Manufacturers

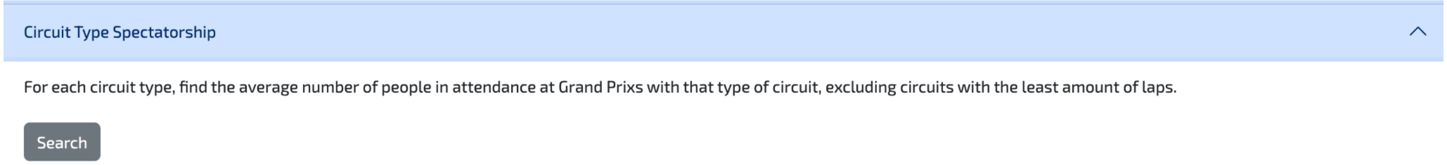
Find the engine manufacturers with more than 5 championship wins.

Table: Constructor

ENGINE	SUM(NUMBEROFWINS)
Mercedes	37
Ferrari	20
Renault	22
Red Bull Powertrains - Honda	6

Nested Aggregation with Group By:

Before:



Circuit Type Spectatorship

For each circuit type, find the average number of people in attendance at Grand Prixs with that type of circuit, excluding circuits with the least amount of laps.

After:

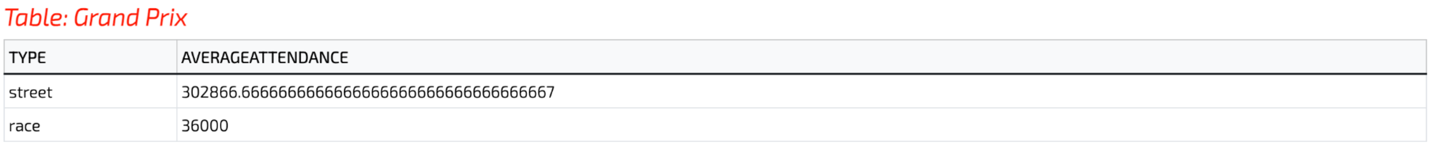
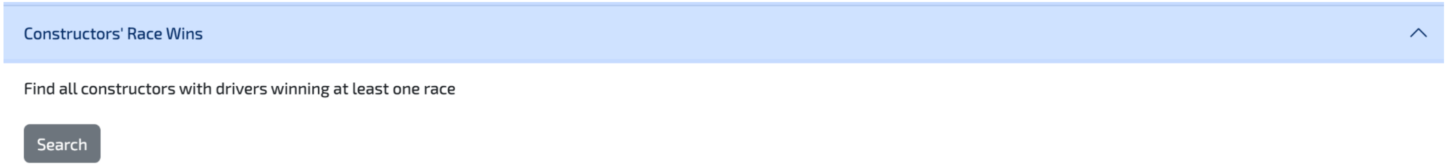


Table: Grand Prix

TYPE	AVERAGEATTENDANCE
street	302866.66666666666666666666666666666667
race	36000

Division:

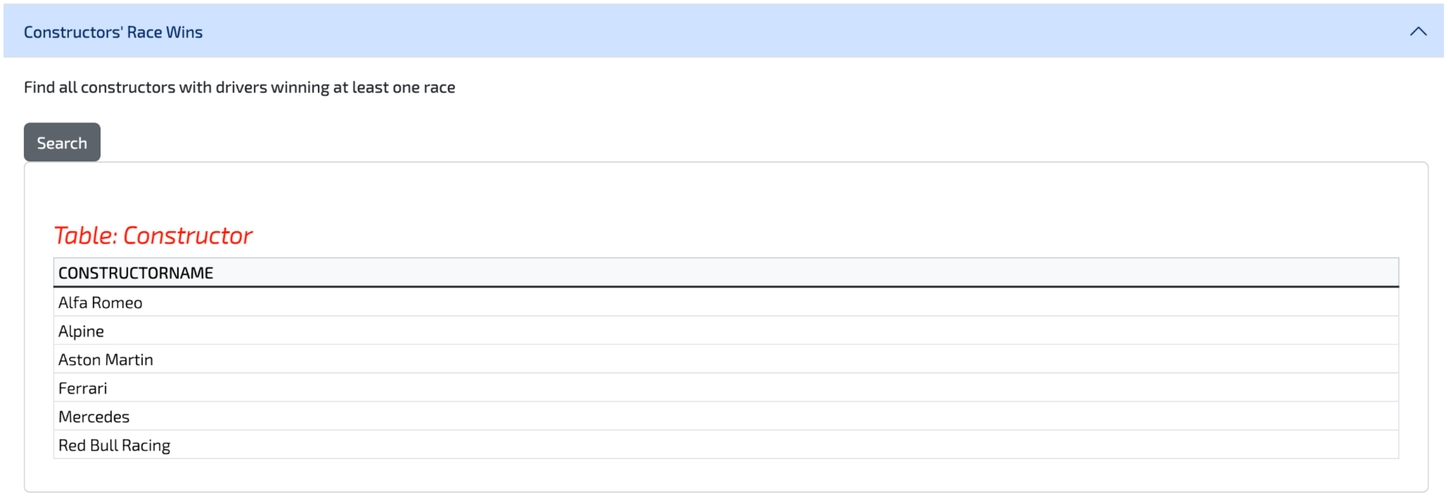
Before:



Constructors' Race Wins

Find all constructors with drivers winning at least one race

After:



Constructors' Race Wins

Find all constructors with drivers winning at least one race

Table: Constructor

CONSTRUCTORNAME
Alfa Romeo
Alpine
Aston Martin
Ferrari
Mercedes
Red Bull Racing