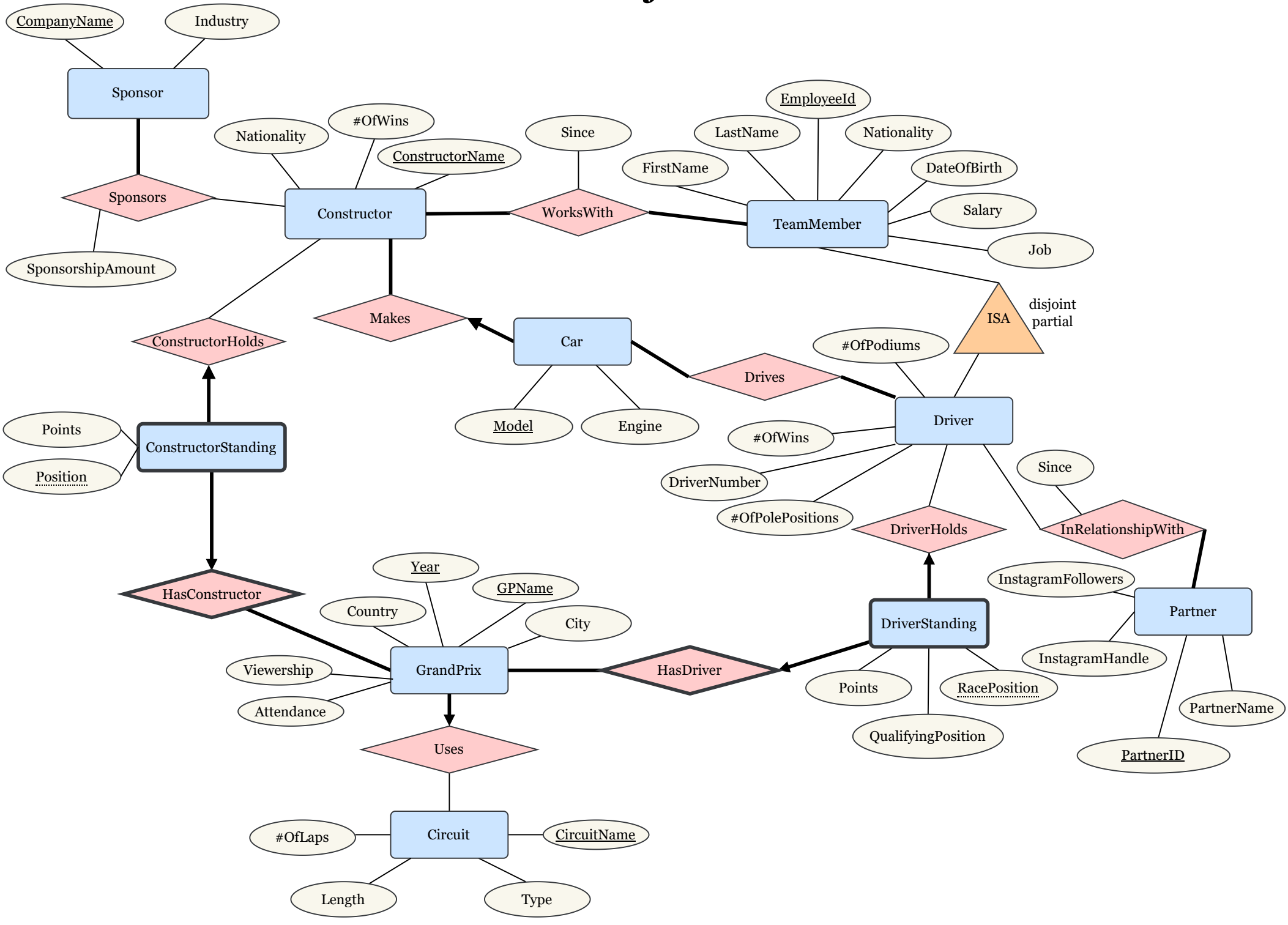
University of British Columbia, Vancouver
Department of Computer Science
CPSC 304 Project - The Formula for Success
Milestone #: 2
Date: October 19, 2023
Group Number: 51

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

"The Formula for Success"



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**)

Decomposition of Circuit Relation

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

Resultant relations: Circuit_Ref <u>(#OfLaps</u>, length), Circuit_2(<u>circuitName</u>, **#OfLaps**, type),

Decomposition of GrandPrix_ConstructorStanding

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

Resultant relations: GrandPrix_ConstructorStanding_2(**<u>position</u>**, **<u>gpName, year</u>**), GrandPrix_ConstructorStanding_Ref(<u>position</u>, points)

Decomposition of GrandPrix_DriverStanding

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

Resultant relations: GrandPrix_DriverStanding_2(**<u>racePosition</u>**, **<u>gpName, year,</u>** qualifyingPosition), GrandPrix_DriverStanding_Ref(<u>racePosition</u>, points)

Normalized Resulting Relations

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

1. Sponsor(<u>companyName: varchar</u>, industry: varchar)
2. Constructor(<u>constructorName: varchar</u>, nationality: varchar, #OfWins: integer)
    - NOT NULL: nationality, #OfWins
    - DEFAULT 0: #OfWins
3. TeamMember(<u>employeeId: integer</u>, firstName: varchar, lastName: varchar, nationality: varchar, dateOfBirth: date, salary: integer, job: varchar)
    - NOT NULL: firstName, lastName, nationality, dateOfBirth, job
4. Car(<u>model: varchar</u>, engine: varchar, **constructorName: varchar**)
    - NOT NULL: constructorName
5. Partner_Ref(<u>partnerId: integer</u>, 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(**<u>instagramHandle: varchar</u>**, instagramFollowers: integer)
7. GrandPrix_Ref(<u>year: year</u>, **<u>circuitName: varchar</u>**, viewership: integer)
8. GrandPrix_2(**<u>circuitName: varchar</u>**, city: varchar)
    - NOT NULL: city
9. GrandPrix_3(**<u>circuitName: varchar</u>**, 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_Ref(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_2(**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_Ref(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_Ref(year, circuitName)
            ON DELETE NO ACTION
);
```

12. Circuit_Ref(#OfLapsr, 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 (
      position                int               PRIMARY KEY,
      points                  int               DEFAULT 0 NOT NULL
);
```

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

```
CREATE TABLE GrandPrix_ConstructorStanding (
      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_2 (
      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_Ref (
      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(**racePosition: integer, gpName: varchar, year: year, employeeID: integer**)
    ***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
            ON DELETE CASCADE
);
```

INSERT Statements:

1. Sponsor(CompanyName: varchar, 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_Red (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’);
```