

# Formel1 Datenbank Projekt

## Inhalt

|   |   |
|---|---|
| Kurzbeschreibung.....                   | 2 |
| ER-Diagramm.....                        | 2 |
| Relationales Modell.....                | 3 |
| Beschreibung der Tests/Funktionen ..... | 4 |
| Trigger .....                           | 4 |
| Procedure .....                         | 5 |
| Function.....                           | 6 |

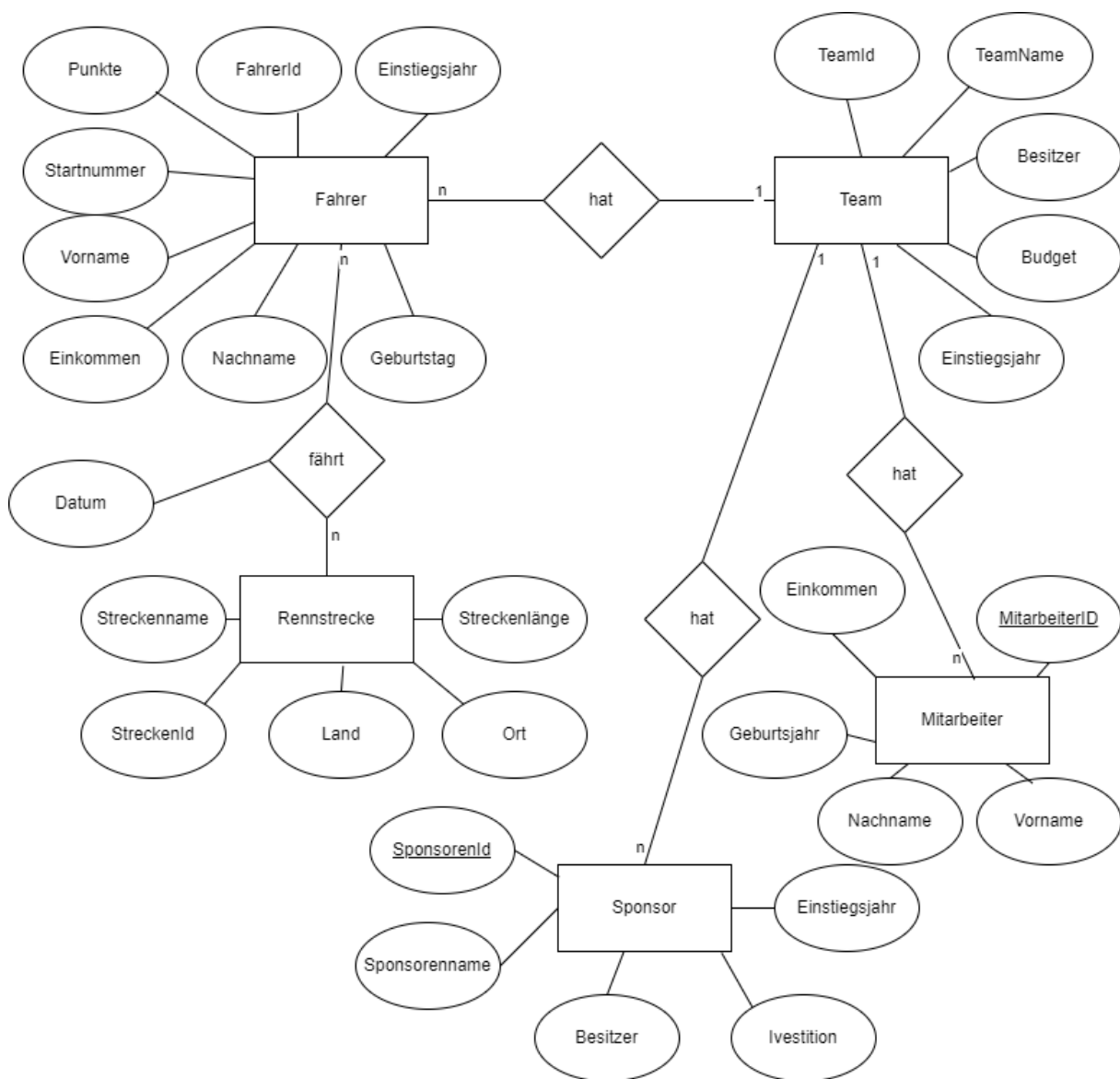


## Kurzbeschreibung

Bei unserem Projekt geht es um das Thema Formel 1. Bei der Formel 1 geht es darum dass verschiedene Teams mit verschiedenen Fahrern an Autorennen teilnehmen und versuchen ein möglichst gutes Ergebnis zu erzielen. Am ende des Jahres gibt es dann einen Gesamtsieger. Bei unserer Datenbank haben wir erfasst welche Teams es in der Formel 1 gibt, welche Fahrer zu diesen Teams gehören und welche Rennstrecken in einem Jahr gefahren werden. Wir haben ebenfalls erfasst was für Sponsoren die einzelnen Teams haben und haben auch die wichtigsten Mitarbeiter der Teams erfasst. Zu denn einzelnen Gebieten haben wir Daten erfasst wie z.B. Gehalt der Fahrer, wie viele Punkte sie im letzten Jahr erzielt haben und noch vieles mehr.

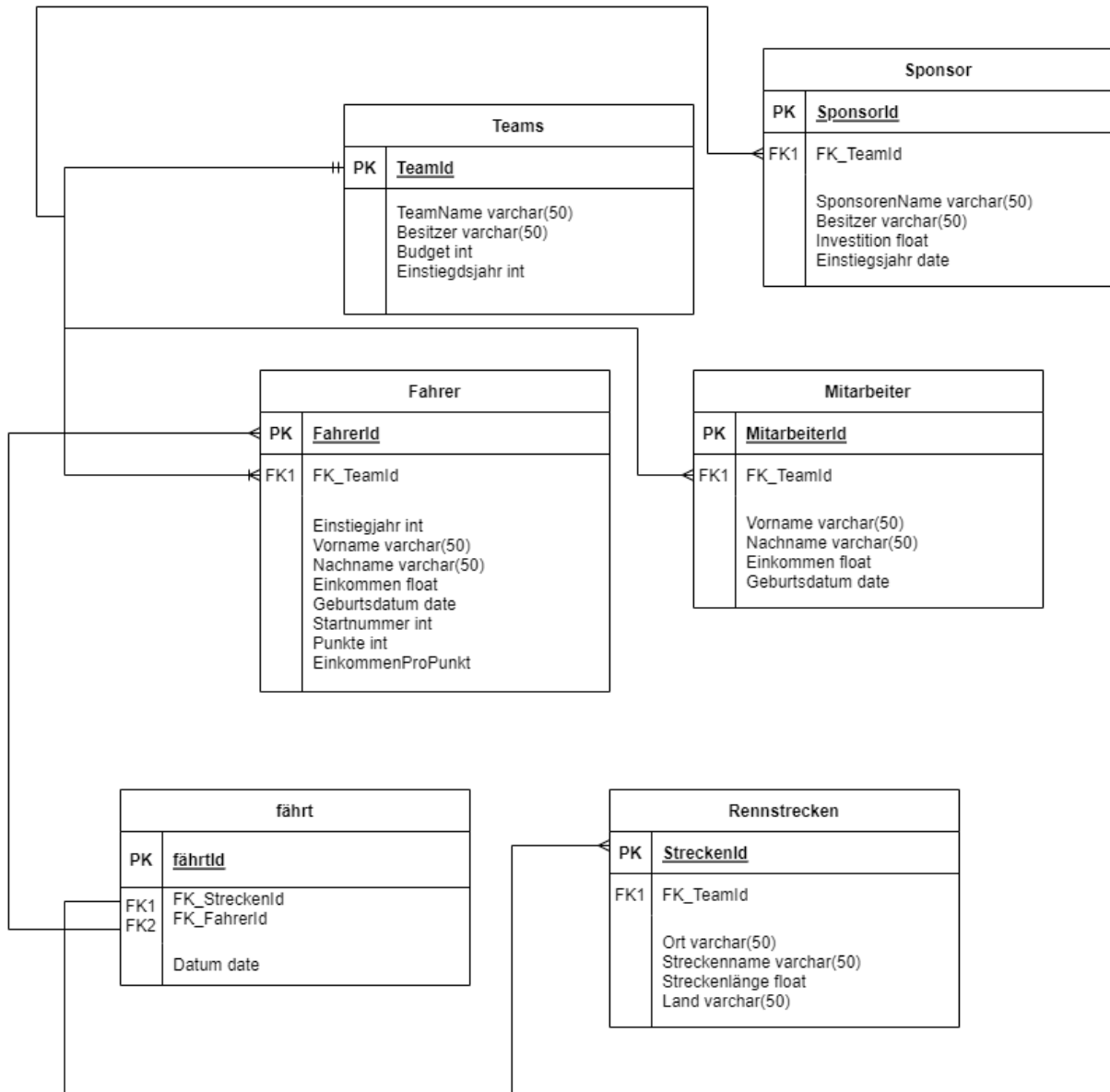
## ER-Diagramm

[https://github.com/LianEifert/M153-Project/blob/main/ER\\_Diagramm\\_Formel1.png](https://github.com/LianEifert/M153-Project/blob/main/ER_Diagramm_Formel1.png)



## Relationales Modell

[https://github.com/LianEifert/M153-Project/blob/main/Relationales Model.png](https://github.com/LianEifert/M153-Project/blob/main/Relationales%20Model.png)



## Beschreibung der Tests/Funktionen

### Trigger

<https://github.com/LianEifert/M153-Project/blob/main/SQL-Files/Trigger.sql>

Der oben gegebene Trigger "GetEinkommenProPunkt" wird ausgelöst, wenn eine neue Zeile in der Tabelle "fahrer" eingefügt wird. Der Trigger führt dann ein UPDATE-Statement auf die Tabelle "fahrer" aus und berechnet den Wert "EinkommenProPunkt" für jede neu eingefügte Zeile.

Die Berechnung von "EinkommenProPunkt" wird durchgeführt, indem der Wert "Einkommen" durch den Wert "Punkte" geteilt wird und das Ergebnis auf zwei Nachkommastellen gerundet wird. Wenn entweder "Einkommen" oder "Punkte" NULL oder 0 sind, wird "EinkommenProPunkt" auf NULL gesetzt.

Die gegebenen Testfälle testen das Verhalten des Triggers in verschiedenen Szenarien.

Im ersten Testfall werden gültige Daten eingefügt, die Berechnung von "EinkommenProPunkt" sollte also erfolgreich durchgeführt werden.

Im zweiten Testfall wird "Einkommen" als NULL eingefügt. In diesem Fall sollte der Trigger "EinkommenProPunkt" auf NULL setzen.

Im dritten Testfall wird "Punkte" als 0 eingefügt. In diesem Fall sollte der Trigger "EinkommenProPunkt" auf NULL setzen, um eine DivideByZero-Exception zu vermeiden.

## Procedure

<https://github.com/LianEifert/M153-Project/blob/main/SQL-Files/Procedure.sql>

Die oben gegebene gespeicherte Prozedur "SelectTeamVonFahrerVorname" nimmt einen VARCHAR-Wert @Fahrervorname als Eingabe und gibt den Namen und das Team eines Fahrers zurück, der den gegebenen Vornamen hat.

Die Prozedur führt zunächst eine Überprüfung durch, ob es einen Fahrer mit dem gegebenen Vornamen gibt. Wenn dies nicht der Fall ist, werden NULL-Werte für "Name" und "TeamName" zurückgegeben.

Wenn @Fahrervorname NULL ist, werden ebenfalls NULL-Werte für "Name" und "TeamName" zurückgegeben.

Wenn keine der oben genannten Bedingungen erfüllt ist, führt die Prozedur eine JOIN-Operation auf den Fahrer- und Team-Tabellen aus und gibt die Fahrerinformationen zurück.

Die gegebenen Testfälle testen das Verhalten der Prozedur in verschiedenen Szenarien.

Im ersten Testfall wird ein gültiger Vorname übergeben. Die Prozedur sollte den Namen und das Team des Fahrers mit diesem Vornamen zurückgeben.

Im zweiten Testfall wird ein ungültiger Vorname übergeben. Die Prozedur sollte NULL-Werte für "Name" und "TeamName" zurückgeben.

Im dritten Testfall wird NULL übergeben. Die Prozedur sollte NULL-Werte für "Name" und "TeamName" zurückgeben.

## Function

<https://github.com/LianEifert/M153-Project/blob/main/SQL-Files/Function.sql>

Die oben gegebene Funktion "durchschnittliches\_einkommen\_pro\_Team" nimmt eine INT-Variable @team\_id als Eingabe und gibt das durchschnittliche Einkommen aller Fahrer im Team mit der gegebenen ID zurück.

Die Funktion führt zunächst eine Überprüfung durch, ob es ein Team mit der gegebenen ID gibt. Wenn dies nicht der Fall ist, wird NULL zurückgegeben.

Wenn @team\_id NULL ist, wird ebenfalls NULL zurückgegeben.

Wenn keine der oben genannten Bedingungen erfüllt ist, führt die Funktion eine Aggregatfunktion AVG auf der Tabelle "Fahrer" aus und gibt das durchschnittliche Einkommen aller Fahrer im Team mit der gegebenen ID zurück.

Die gegebenen Testfälle testen das Verhalten der Funktion in verschiedenen Szenarien.

Im ersten Testfall wird eine gültige ID übergeben. Die Funktion sollte das durchschnittliche Einkommen des Teams mit dieser ID berechnen.

Im zweiten Testfall wird eine ungültige ID übergeben. Die Funktion sollte NULL zurückgeben.

Im dritten Testfall wird NULL übergeben. Die Funktion sollte NULL zurückgeben.