Alexandra Speer
Nicolas Henzel
Gruppe 2
Hochschule der Medien
Sommersemester 23

HOCHSCHULE
DER MEDIEN

Data Warehouse Workshop

Prof. Dr. Klaus Freyburger

Inhaltsverzeichnis

Daten

Die Datensätze stammen von Kaggle[1]. Inhalt sind die Weltmeisterschaften der Formel 1 im Zeitraum von 1950 bis 2023. Die Daten sind durch die Ergast Developer API[2] zusammengestellt worden, die Renndaten für nicht kommerzielle Zwecke zur Verfügung stellt.

Die Formel 1 (oft auch F1) ist eine Formelserie, die durch den Automobil Dachverband Fédération Internationale de l'Automobile (FIA) autorisiert ist. Formelserie bedeutet hierbei, dass bestimmte Regeln (Formeln) auf technischer Ebene für die Leistungsfähigkeit der Fahrzeuge festgelegt wurden, um einen Wettkampf unter gleichbleibenden Bedingungen zu ermöglichen. Die Formel 1 Weltmeisterschaft fand erstmals 1950 statt und besteht pro Saison aus bis zu 23 Grand Prix (französisch für „Großer Preis") Rennen. Dies sind Einzelrennen auf ausgewählten Rennstrecken in jeweils unterschiedlichen Ländern. Dabei sammeln die Fahrer abhängig von ihrer Endposition bei diesen Rennen Punkte. Am Ende der Saison gewinnt der Fahrer mit den meisten Punkten. Außerdem erhalten die Konstrukteure der Wagen Punkte, die ebenfalls am Ende der Saison ausgewertet werden.[3]

Die Daten beinhalten alle Informationen der Formel 1 Weltmeisterschaften seit 1950:
Rennen, Fahrer, Konstrukteure, Qualifizierungen, Rennstrecken, Rundenzeiten, Boxenstopps und Ergebnisse. Diese Informationen sind in 14 Datensätze aufgeteilt. Eine Übersicht der Daten in Python findet sich im DataExploration.ipynb.
Die wichtigsten Erkenntnisse aus dieser Übersicht sind:

- Zellen, bei denen kein sinnvoller Wert eingetragen ist, sind mit \N markiert.
  Das wird im Data Wrangling behoben
- Die Daten sind bereits „tidy":
  Jede Zeile beinhaltet eine Beobachtung,
  jede Spalte beschreibt eine Variable,
  in jeder Zelle ist genau ein Wert (und nicht mehrere).[4]
- Die Datensätze besitzen mindestens eine Id Spalte (Primärschlüssel). wodurch sich die Daten leicht referenzieren lassen. Manche Datensätze haben weitere Id Spalten aufgelistet, die als Fremdschlüssel verwendet werden können.

[1] https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020

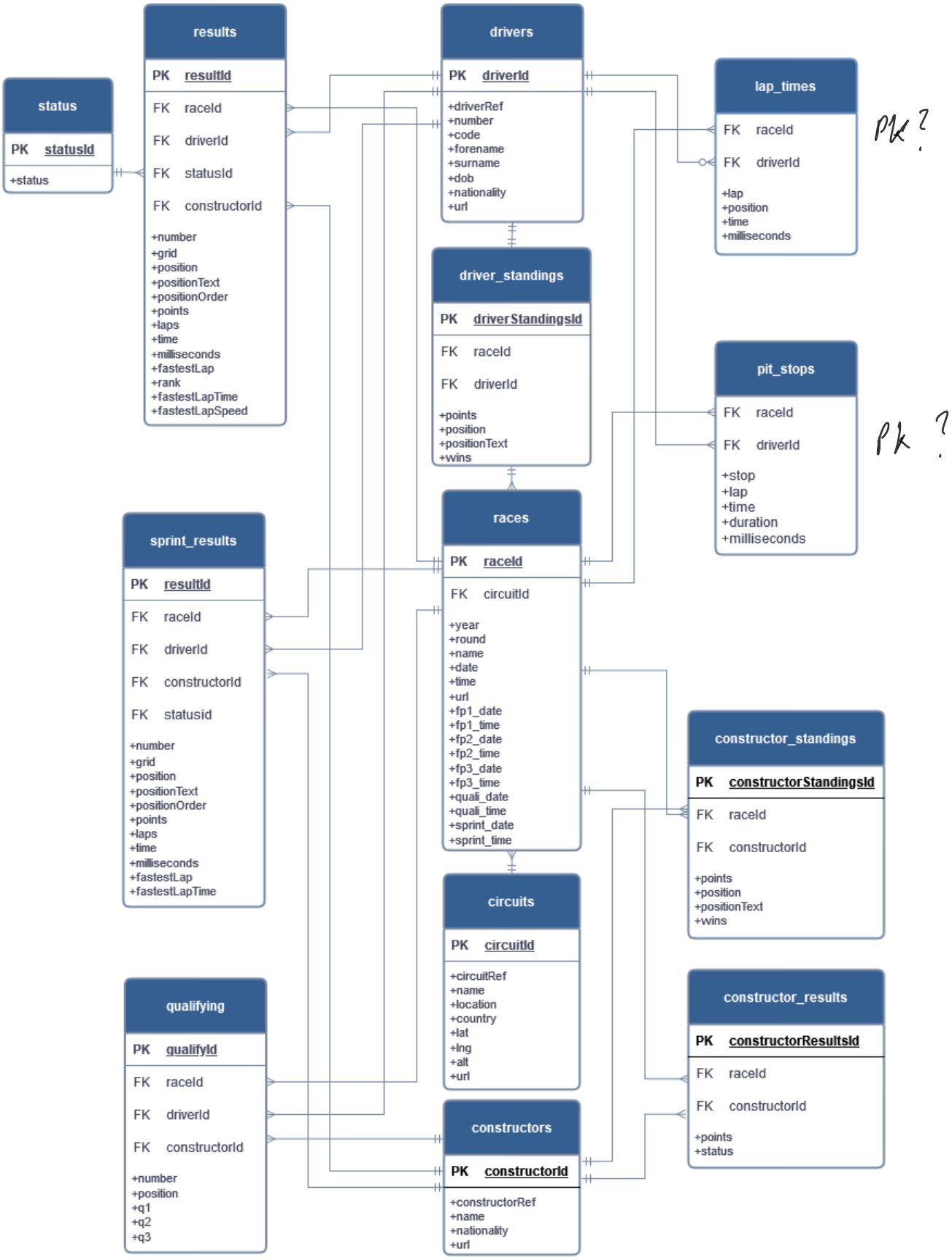
[2] http://ergast.com/mrd/

[3] https://de.wikipedia.org/wiki/Formel_1

[4] https://cran.r-project.org/web/packages/tidyr/vignettes/tidy-data.html

Konzeptionelles Modell

eigentlich schon logisch



Design der Umsetzung

Logisches Modell

Offen - Wird bei der Implementierung erstellt.

Data Wrangling

Die Daten wurden im ersten Schritt im DataExploration.ipynb untersucht und in einer flachen Form zusammengeführt. Dabei haben wir uns auf die Fahrer Daten fokussiert und die Id Spalten genutzt, um die unterschiedlichen Datensätze miteinander zu joinen. Die Konstrukteur Daten sind im ER-Modell enthalten. Die Struktur dient zur ersten Übersicht, sowie beispielhaften Verbindung der Datensätze zu einem finalen Datensatz. Damit wurden zusätzlich Auswertungen beispielhaft im Python Skript umgesetzt.

Die Datensätze, die genutzt wurden, sind:

- circuits.csv
- driver_standings.csv
- drivers.csv
- lap_times.csv
- pit_stops.csv
- results.csv
- races.csv
- status.csv
- constructors.csv
- constructor_standings.csv
- constructor_results.csv
- qualifying.csv
- sprint_results.csv

Der seasons.csv Datensatz wurde nicht genutzt, da hier keine neuen Informationen enthalten sind.

Zum Laden der Daten in DWC wurde eine JSON-Datei erstellt (180_FormulaOne.json), die die Tabellen und Verbindungen anlegt.
Folgende Anpassungen haben wir vorgenommen:

- Driver_standings.csv - positionText Spalte entfernt
- Lap_times.csv - Neuen Datensatz lap_times_grouped.csv erstellt, der gruppierte Werte enthält

```
# group laptimes of drivers for each different race by taking the mean over
all laps of a driver for specific race
# milliseconds could be converted into minutes:seconds.milliseconds format if
needed
df_grouped_lap_times = df_lap_times.groupby(
['driverId','raceId']
)['milliseconds'].mean().reset_index()
```

```
# rename column for more clarity
df_grouped_lap_times.rename(
{'milliseconds':'avgMillisecondsLap'},axis=1, inplace= True
)
df_grouped_lap_times.to_csv("Data/lap_times_grouped.csv", index=True)
```

- Pit_stops.csv - duration Spalte entfernt
- Results.csv - positon, positionText Spalte entfernt
- Constructor_standings.csv - positionText Spalte entfernt
- Sprint_results.csv - positon, positionText Spalte entfernt
- Alle „URL“ Spalten in jedem Datensatz entfernt (für die Auswertung nicht relevant)
- Alle Spalten mit Datum auf das Format YYYY-MM-DD angepasst
- Werte in allen mit \N markierten Zellen werden entfernt, damit beim Import in DWC korrekt NULL eingetragen werden kann

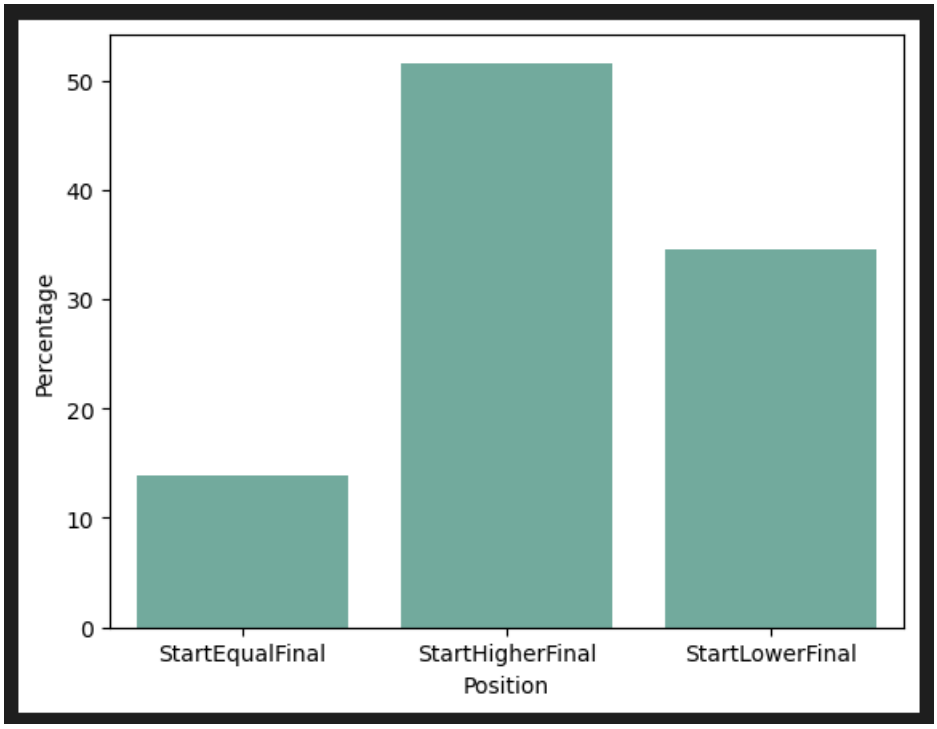
Die konkreten Schritte sind im DataWrangling.ipynb nachzuvollziehen.

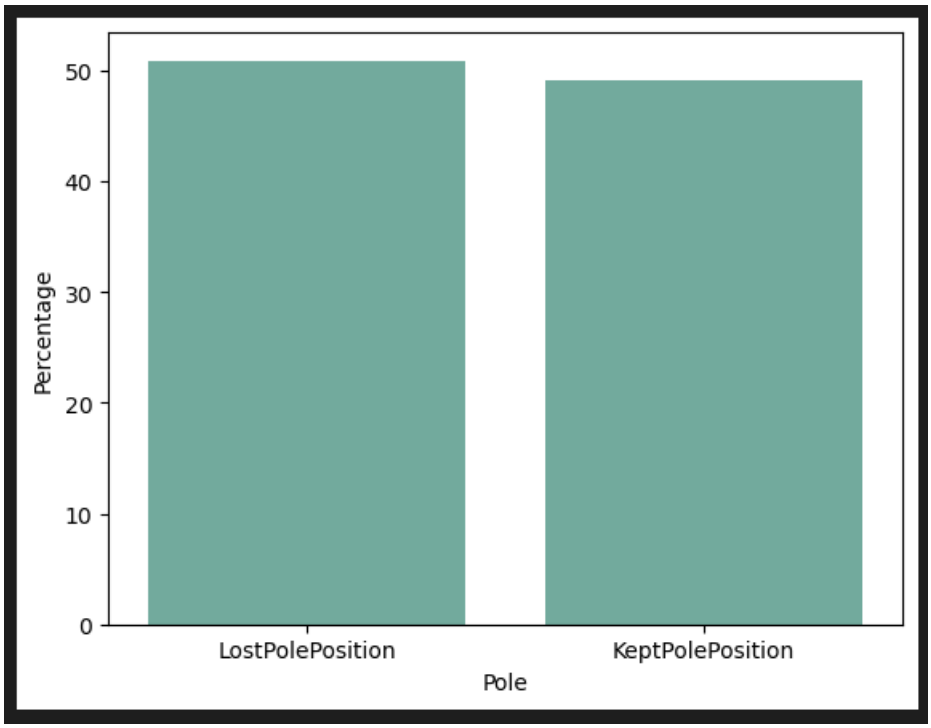
Die beispielhaften Auswertungen werden im Anschluss erläutert.

Beispielhafte Auswertungen

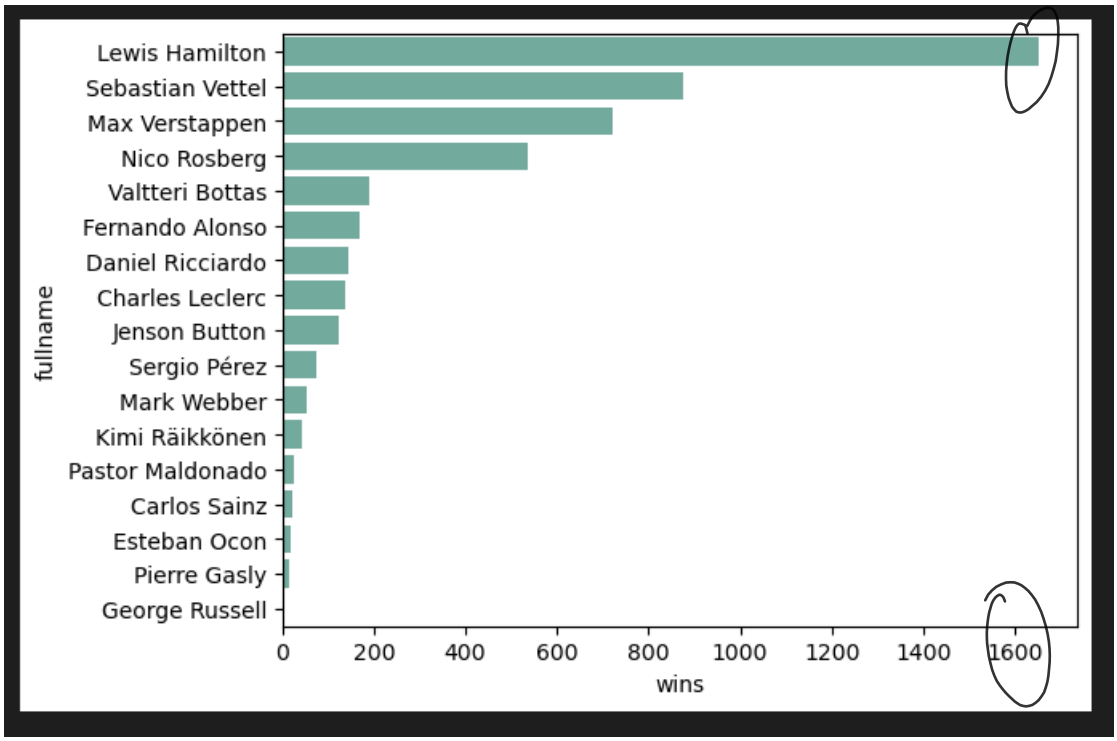
Vorschläge zur Auswertung finden sich in der DataExploration.ipynb Datei.
Wir wollen damit folgende Fragen beantworten:

- Wie ist das Verhältnis zwischen Start und Endposition?





- Welcher Fahrer hat die meisten Grand Prix Rennen gewonnen?



plausibel?

- Welcher Hersteller hat die meisten Weltmeisterschaften gewonnen?
- Welcher Fahrer hat die meisten Weltmeisterschaften gewonnen?
- In welcher Saison sind die meisten Unfälle passiert?
- Welcher Hersteller hat die meisten Defekte / Ausfälle / Unfälle?
- Welche Strecken werden mit den kürzesten, welche mit den längsten Zeiten gefahren?

Änderung der Zeiten im Zeitablauf?

Nationalität / Land Strecke, Fahrer, Konstrukteur

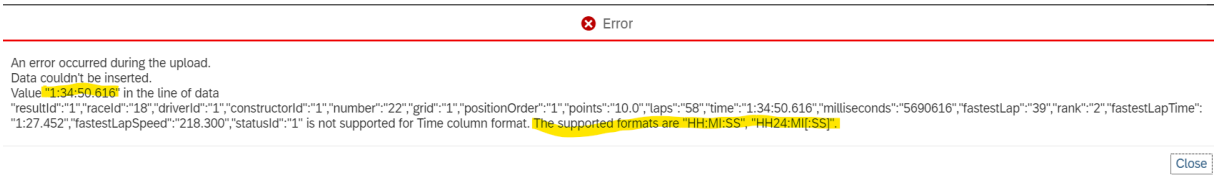
Durchschn. → Korrelation Ergebnis?

Rangfolge der Rundenzeiten = Ergebnis des Rennens?

Implementierung

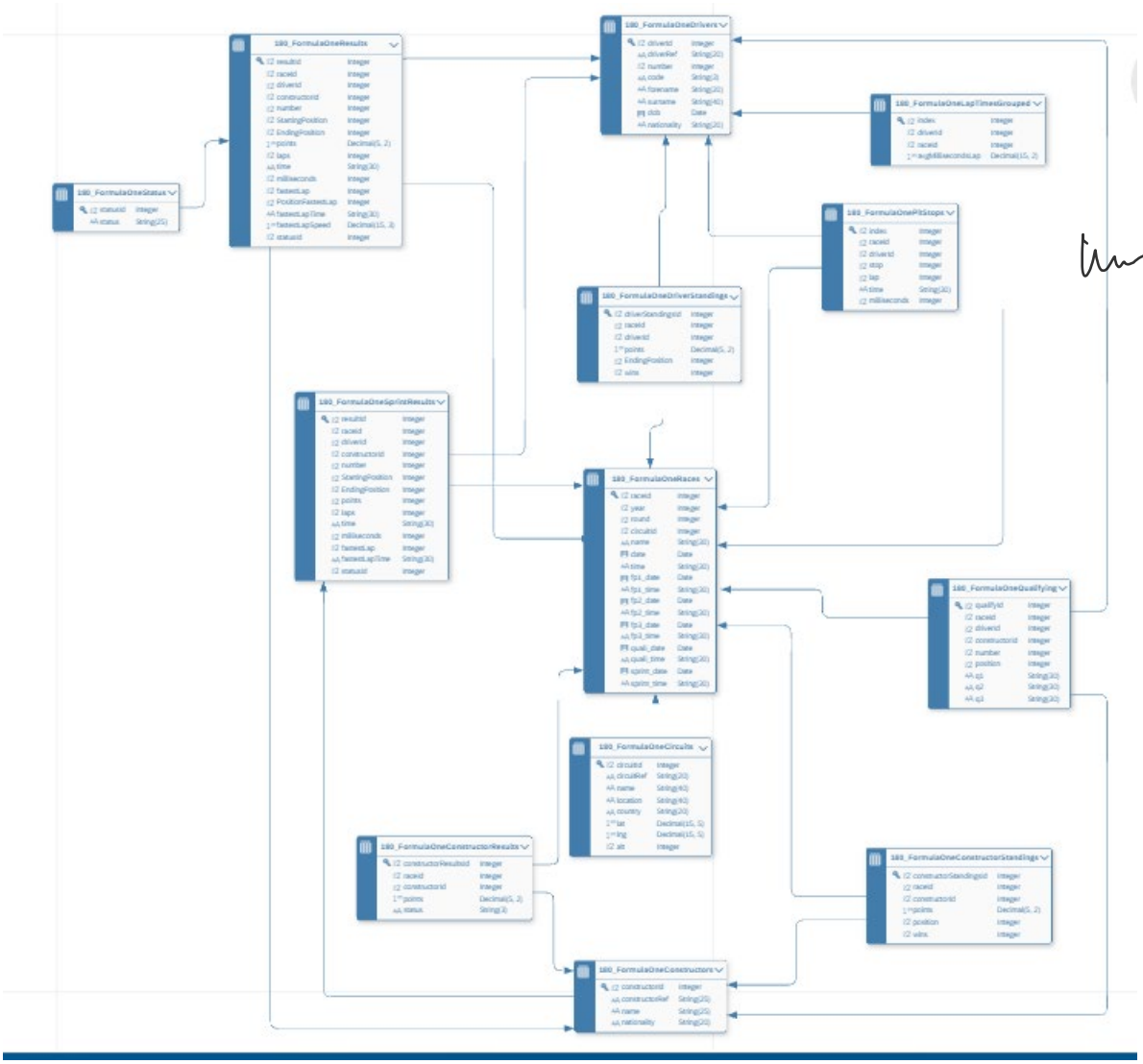
Die Implementierung der Tabellen wurde mit der erstellten JSON-Datei durchgeführt, wodurch das E/R Modell per Import in DWC angelegt wurde.

Die erste Herausforderung war, dass die Datumsformate in der richtigen Form sein mussten, wir haben YYYY-MM-DD gewählt. Spalten, die einen Zeitwert enthalten haben, wurden nicht korrekt erkannt, da diese nicht im H:MM:SS Format vorliegen, sondern H:MM:SS.MS.



Aus diesem Grund haben wir diese Spalten als String importiert, nachdem die Fehlermeldung in DWC aufgetreten ist. Die Zeitwerte sind als Millisekunden in den entsprechenden Spalten in Integer Werten vorhanden, um dort Auswertungen machen zu können.
Außerdem haben wir bemerkt, dass einige Zeichenketten länger waren als vermutet, wodurch die Tabellen händisch angepasst werden mussten (z.B. surname Spalte in Drivers Tabelle, kommt mit 20 Zeichen nicht aus, weshalb die Länge auf 40 Zeichen erhöht wurde). Zusätzlich gibt es Spalten mit Zahlen, die auf den ersten Blick wie Integers aussehen, nach dem Import jedoch Float sind, was wir auch anpassen mussten (z.B. points Spalte in results.csv).

Unser E/R Modell in DWC sieht folgendermaßen aus (180_FormularOne):



Anmerkungen für Vorgehen

Nächste Schritte:

- Views bauen, 3 finale Dimensionen – Fahrer, Rennen, Hersteller
- Stories bauen, um Auswertungen darzustellen – iterative Anpassungen an Datengrundlage, um Auswertungen korrekt abzubilden
- Logisches Modell abbilden

Wir bitten um Feedback per Zoom.

Zeitlich sind wir folgendermaßen verfügbar:

- Täglich ab 16 Uhr, außer am 5. April

Fragen von unserer Seite:

- Können wir Hierarchien abbilden mit Fremdschlüsseln?
  Aus unserer Sicht sind in den Daten keine Hierarchien abgebildete auf Attribut-ebene. Möglich wäre bspw. eine Hierarchie im Results Datensatz auf RaceId und DriverId
- Die Aggregation der Tabelle Lap Times ist in Python erstellt worden. Ist das so in Ordnung, oder soll diese in DWC durchgeführt werden?
  Wenn es in DWC erstellt werden soll, wie ist das am besten möglich?

wie wäre es mit Nationalität / Land?

Fahrer, Strecke, Konstrukteur

1.) Nur wenn das Datenvolumen dies erfordert hier 15 MB → in DWC

2.) Beides möglich View oder Dataflow

Eigenschaft der Measures

Aggregationsmodus oder Python Script

nur wenn nicht mit Standardmitteln möglich

Quellen

https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020
Zuletzt besucht am 30.03.2023 19 Uhr

http://ergast.com/mrd/
Zuletzt besucht am 30.03.2023 19 Uhr

https://de.wikipedia.org/wiki/Formel_1
Zuletzt besucht am 30.03.2023 19 Uhr

https://cran.r-project.org/web/packages/tidyr/vignettes/tidy-data.html
Zuletzt besucht am 30.03.2023 19 Uhr