Analyzing Ferrari's last years in F1

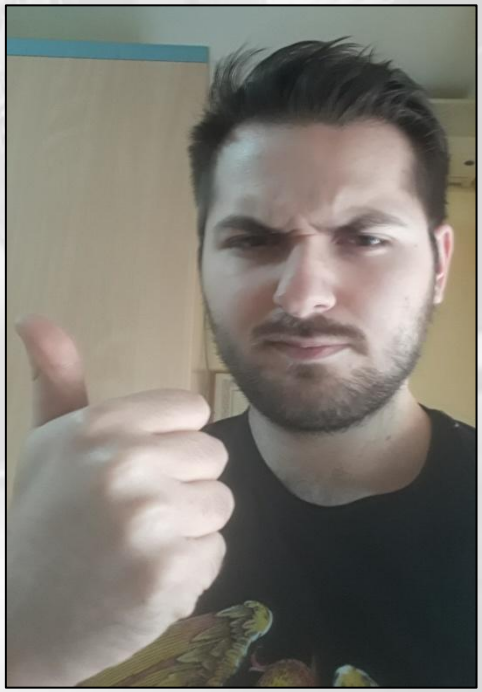
- Talend portion -

Luca Sannino 1542194

- Hi, my name is Luca… and I've recently become the new team principal at Ferrari!

2014-2019

2019-2020



2020-????

- As a team principal, it is my duty to keep improving the team so that we can compete with the other constructors and, hopefully, win the season.

- However, before I can improve something, I first need to understand WHAT it needs to be improved!

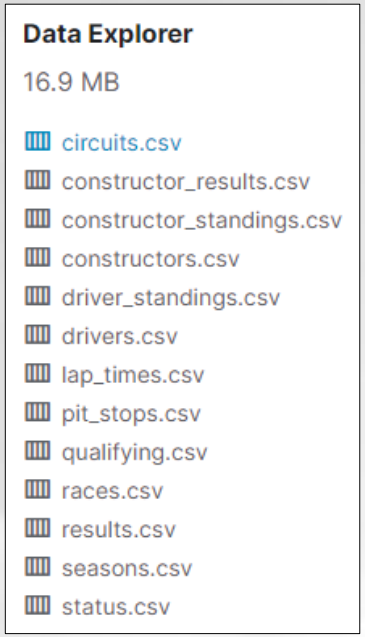
• So I've selected three areas where I feel the team is not meeting my expectations.

POTENTIAL ISSUES

1 – Our pit stops are subpar
2 – Vettel is underperforming
3 – The car is slower than Mercedes'

• The objective of this project is to change that "*where I feel*" to "*where I know*". How?
• By using a graph database to query data obtained after an integration process.
• This portion will focus on the integration!

- My instruments for this portion will be:



A series of .csv files containing stats from the entire F1 history

Talend Data Integration: an open-source software that allows the user to perform ETL operations

POTENTIAL ISSUE 1

Are our pit stops subpar?

• Last year Red Bull registered the quickest pit stop in the entire history of Formula 1.

WHAT I WANT TO KNOW

Was it just luck or a symptom of a bigger problem: that we are just slower?

- The objective of this exercise will be to collect all of our pit stops from last year by integrating the .csv files, and then using the output to calculate the average pit stop time per pilot. After doing the same for Red Bull, we will compare the results and see how we stack up.
- Note: the portion on the graph database will use more generic global mappings, but here I wanted more complex queries in order to showcase Talend's functionalites. The same holds true for all the following exercises.
- Let's begin by finding our pit stops by integrating the sources!

SOURCE SCHEMAS

- pitStops(raceId, driverId, stop, lap, time, duration, milliseconds)
- drivers(driverId, driverRef, number, code, forename, surname, dob, nationality, url)
- races(raceId, year, round, circuitId, name, date, time, url)

GLOBAL SCHEMA

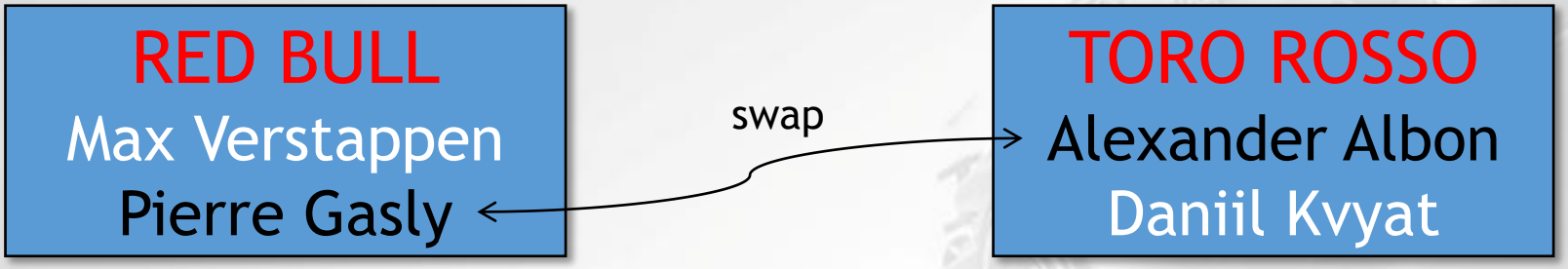
- Ferrari2019PitStops(pilot, race, date, stop, lap, duration)

GAV MAPPING

```
{ (a, b, c, d, e, f) | pitStops(x1, x2, d, e, y1, f, y2),
                       races(x1, 2019, y3, y4, a, b, y5, y6),
                       drivers(x2, y7, y8, y9, a, x3, y10, y11, y12),
                       (x3 == "Vettel" || x3 == "Leclerc") }
                    -> Ferrari2019PitStops(a, b, c, d, e, f)
```

Note: in the sources there's no info about constructors!

• Finding Red Bull pit stops won't be as trivial since there was a mid-season driver swap last year.



• We are not interested in pit stops from Toro Rosso, so we have to make sure that Gasly and Albon were driving for Red Bull before extracting their pit stop times from a specific race!

ADDITIONAL SOURCE SCHEMAS

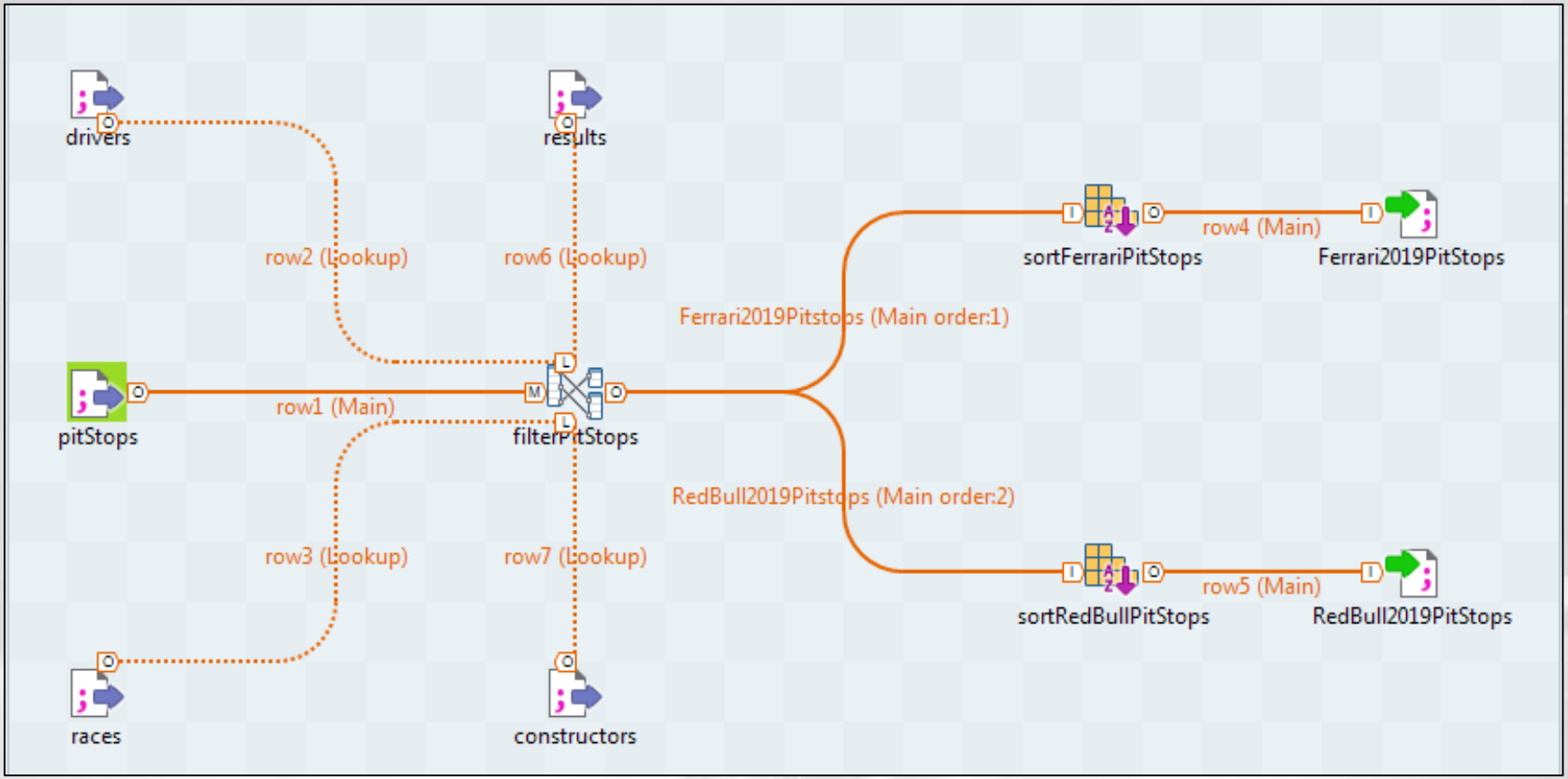
- results(resultId, raceId, driverId, constructorId, number, …)
- constructors(constructorId, constructorRef, name, nationality, url)

GLOBAL SCHEMA

- RedBull2019PitStops(pilot, race, date, stop, lap, duration)

GAV MAPPING

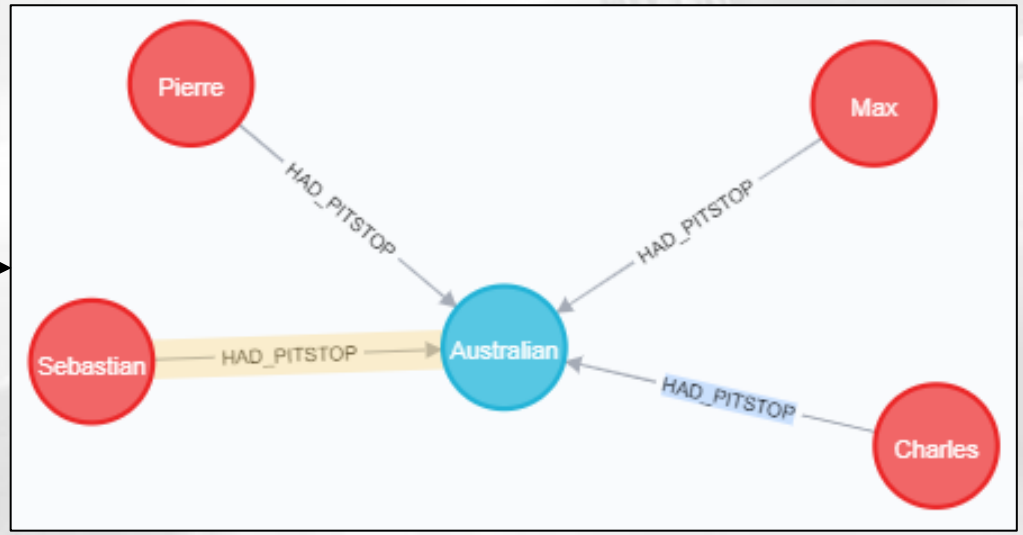
```
{ (a, b, c, d, e, f) | pitStops(x1, x2, d, e, y1, f, y2),
                       races(x1, 2019, y3, y4, a, b, y5, y6),
                       drivers(x2, y7, y8, y9, a, y10, y11, y12, y13),
                       results(y14, x1, x2, x3, y15, …),
                       constructors(x3, y16, “Red Bull”, y17, y18) }
                       -> RedBull2019PitStops(a, b, c, d, e, f)
```



- Now let's see how this works in practice with a live demonstration using Talend!

- Once we have integrated our data, we must query it in order to obtain the averages.
We can do this by inserting our data as a input .csv file to a graph database (like Neo4j) and query the data there!

pilot	race	race_date	stop	lap	duration_in_ms
Max	Australian Grand Prix	17/03/2019	1	25	21157
Pierre	Australian Grand Prix	17/03/2019	1	37	21269

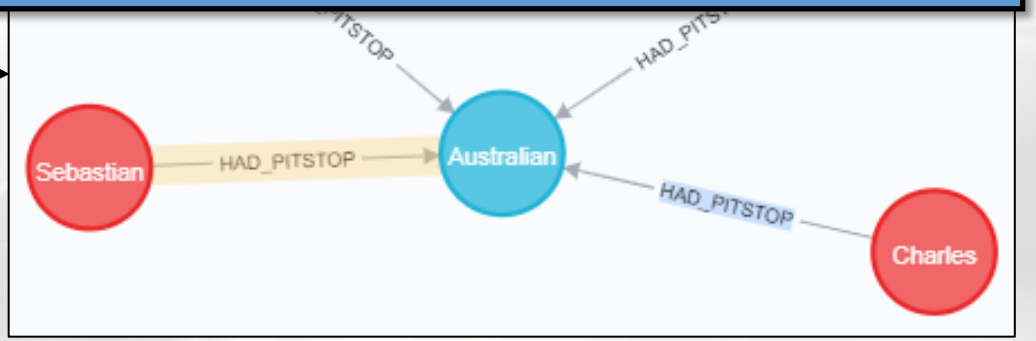


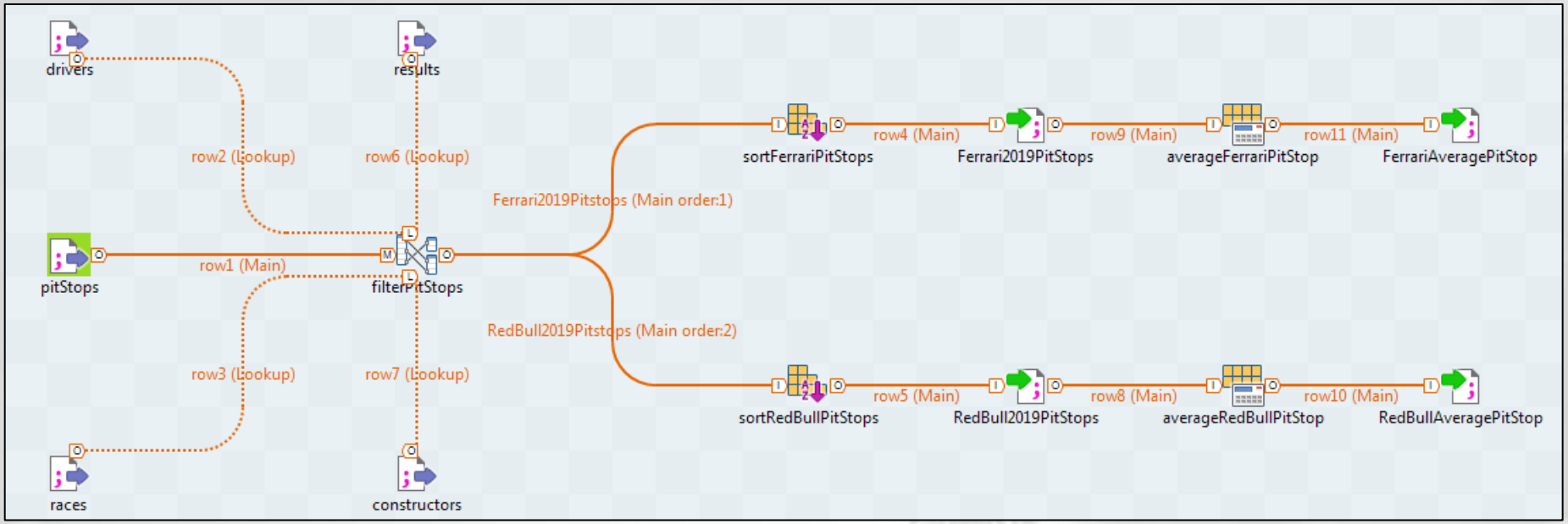
• Once we have integrated our data, we must query it in order to obtain the averages.
We can do this by inserting our data as a input .csv file[illegible]and query t[illegible]

But this is another topic for another time!

Let's see if we can replicate the same results just by using Talend data transformation tools!

pilot	[illegible]	[illegible]ms
Max	[illegible]	
Pierre	[illegible]	





- Now let's see how this works in practice with a live demonstration using Talend!

CONCLUSIONS

pilot	average_duration
Charles	24281
Sebastian	24942

pilot	average_duration
Alexander	25059
Pierre	23657
Max	23682

- Looking at the results, and since Albon raced in too few races to make his average relevant, it's fair to say that our pit stops are slower. We need either to improve or account for a ~1 second delay when making strategies that involve pitting.

POTENTIAL ISSUE 2

Is Vettel underperforming?

• Last year many claimed Vettel was feeling the pressure from his new teammate, Leclerc, which led him to some major mistakes.

WHAT I WANT TO KNOW

Is Vettel in a difficult situation, or are people just blinded by the hype for our younger driver?

• The objective of this exercise will be to collect all the placements of Vettel since he joined F1, alongside the ones of his various teammates, by integrating the .csv files. We'll then use the output to calculate how many times he has placed better than his teammate per season, so that we can see if there was a downgrade in his perfomance across the years.

• Let's begin by collecting the relevant placements!

SOURCE SCHEMAS

- results(resultId, raceId, driverId, constructorId, number, …, position, …)
- drivers(driverId, driverRef, number, code, forename, surname, dob, nationality, url)
- races(raceId, year, round, circuitId, name, date, time, url)
- constructors(constructorId, constructorRef, name, nationality, url)

GLOBAL SCHEMA

- VettelAgainstTeammates(race, date, year, vettel_position, teammate, teammate_position, constructor)

GLOBAL SCHEMA

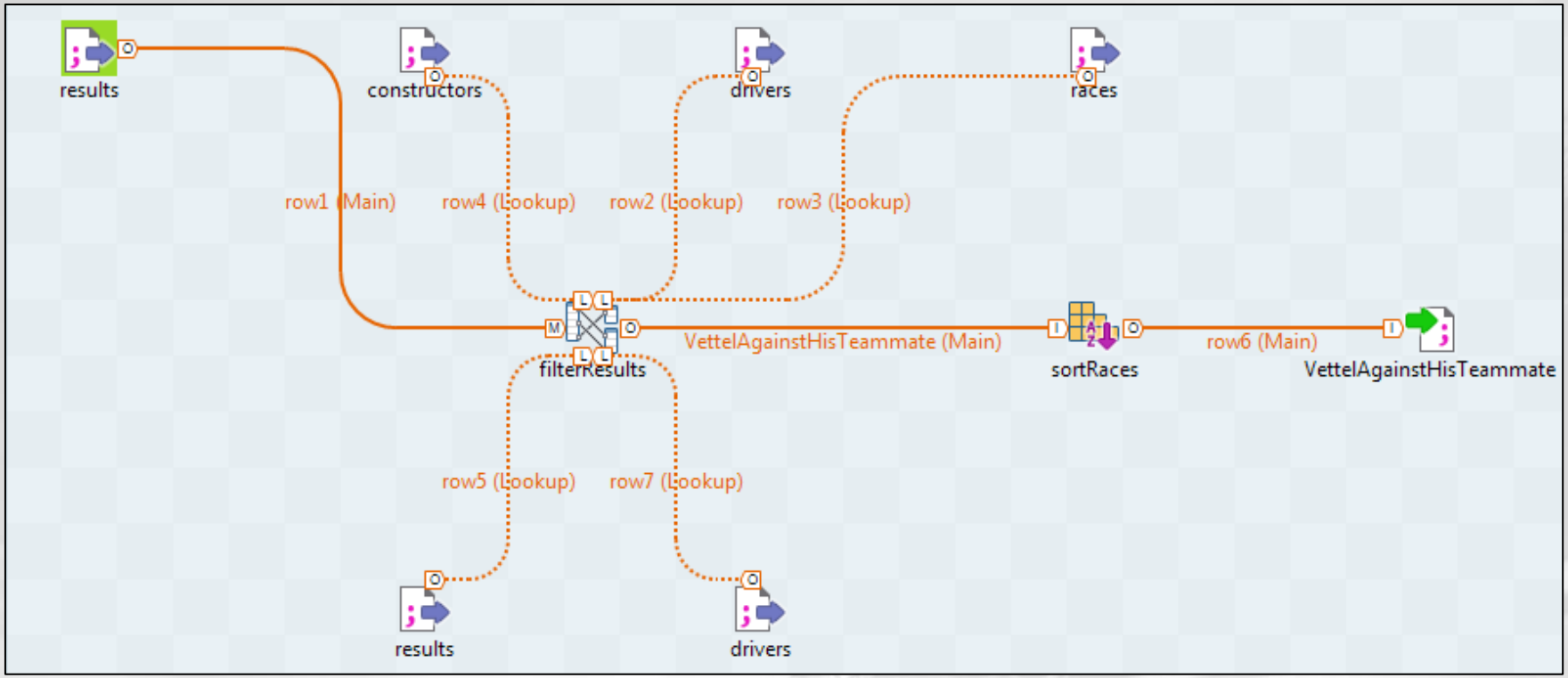
- VettelAgainstTeammates(race, date, year, vettel_position, teammate, teammate_position, constructor)

GAV MAPPING

```
{ (a, b, c, d, e, f, g) | results(y1, x1, x2, x3, y2, …, d, …),
                          races(x1, c, y3, y4, a, b, y5, y6),
                          drivers(x2, y7, y8, y9, y10, “Vettel”, y11,
                                  y12, y13),
                          constructors(x3, y14, g, y15, y16),
                          results(y17, x1, x4, x3, y18, …, f, …),
                          drivers(x4, y19, y20, y21, y22, e, y23,
                                  y24, y25),
                          x2 != x4,
                          c >= 2008 }
                       -> VettelAgainstTeammates(a, b, c, d, e, f, g)
```

The drivers must be different

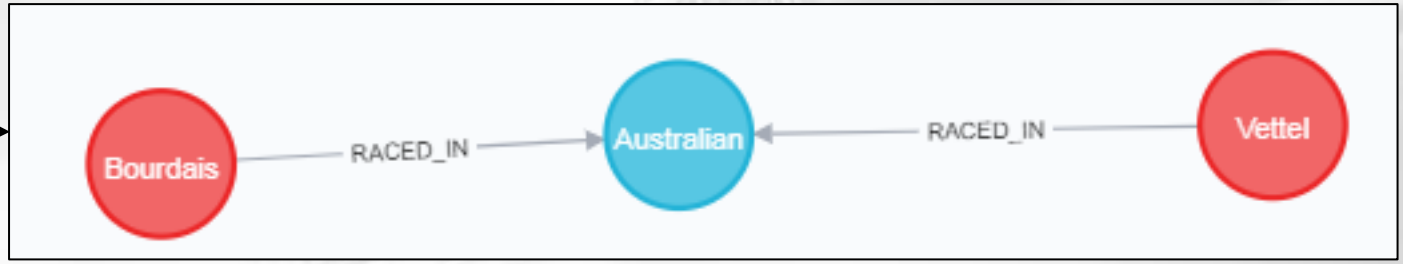
First full season for Vettel



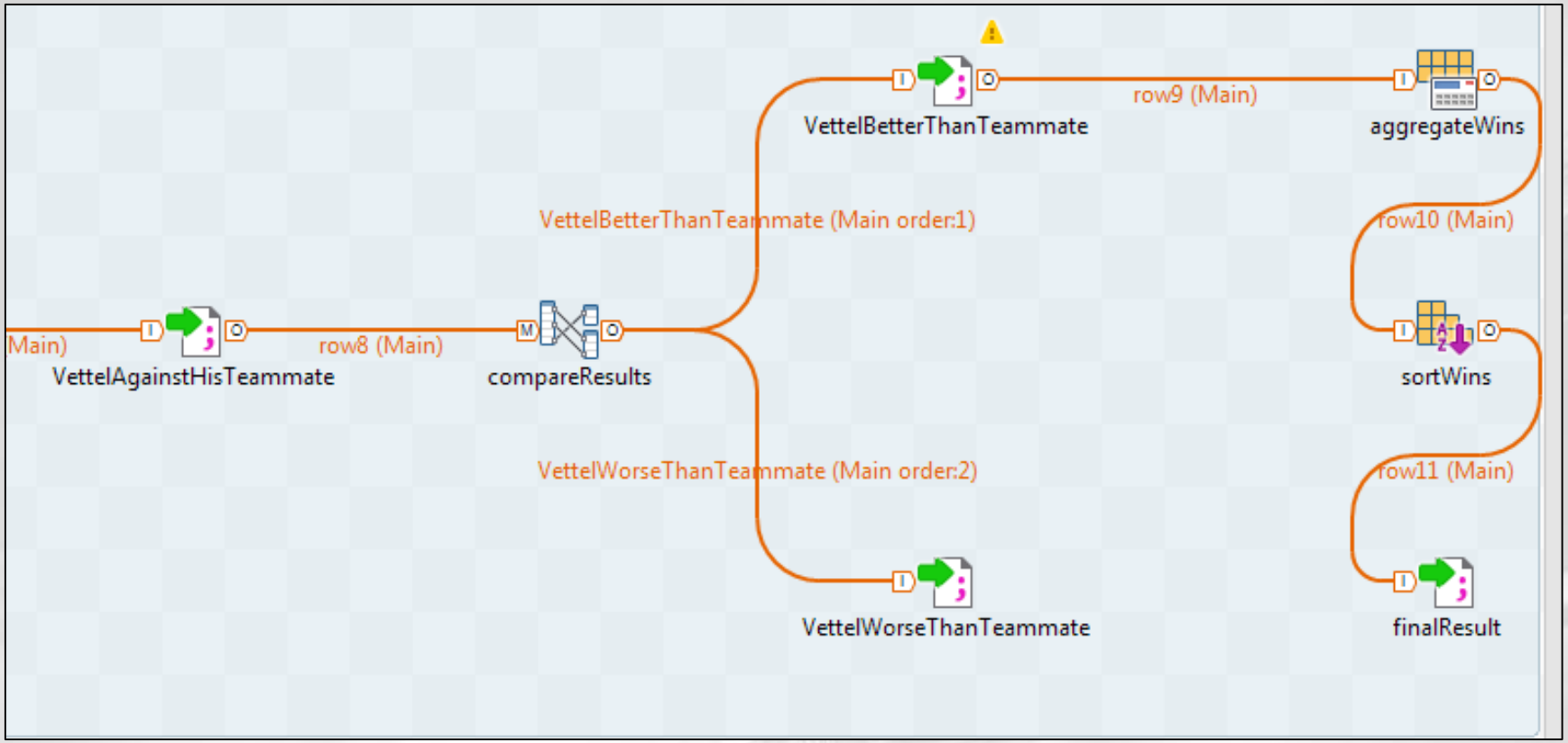
- Now let's see how this works in practice with a live demonstration using Talend!

• After integrating our data, we must query it in order to count how many times Vettel has placed better than his teammate per season. Again, we can use a graph database like Neo4j

race_name	date	year	vettel_position	teammate	teammate_position	constructor
Australian Grand Prix	16/03/2008	2008	\N	Bourdais	7	Toro Rosso



• We can still use Talend's transformation tools however!



- Now let's see how this works in practice with a live demonstration using Talend!

CONCLUSIONS

year	teammate	times_vettel_has_won
2008	Bourdais	12
2009	Webber	8
2010	Webber	11
2011	Webber	16
2012	Webber	13
2013	Webber	18
2014	Ricciardo	5
2015	Räikkönen	14
2016	Räikkönen	14
2017	Räikkönen	16
2018	Räikkönen	12
2019	Leclerc	12

- Even though Vettel had a hard time against Leclerc, it was surprising to learn that it wasn't any better last year or in 2014, where Ricciardo clearly gave a stronger performance!

POTENTIAL ISSUE 3

Are we slower than Mercedes?

- It’s not a secret that Mercedes is dominating the recent era of Formula 1.

2014	Lewis Hamilton [30]	29	Mercedes	Mercedes
2015	Lewis Hamilton [30]	30	Mercedes	Mercedes
2016	Nico Rosberg [33]	31	Mercedes	Mercedes
2017	Lewis Hamilton [30]	32	Mercedes	Mercedes
2018	Lewis Hamilton [30]	33	Mercedes	Mercedes
2019	Lewis Hamilton [30]	34	Mercedes	Mercedes

WHAT I WANT TO KNOW

Is this the result of unlucky circumstances or is our car just slower?

- The objective of this exercise will be to collect all fastest laps during races from both Mercedes and Ferrari since 2014 by integrating the .csv files. We'll then use the output to filter the races where each driver had the opportunity to register a valid fast lap, and then sum the fast lap times grouped by season and pilot, and ultimately compare the results and see who has a quicker time.

- Let's begin by collecting the fastest laps!

SOURCE SCHEMAS

- results(resultId, raceId, driverId, constructorId, number, …, fastestLap, …, fastestLapTime, …)
- drivers(driverId, driverRef, number, code, forename, surname, dob, nationality, url)
- races(raceId, year, round, circuitId, name, date, time, url)
- constructors(constructorId, constructorRef, name, nationality, url)

GLOBAL SCHEMAS

-FerrariFastestLapTimes(race, date, year, driver, lap, time)
-MercedesFastestLapTimes(race, date, year, driver, lap, time)

GLOBAL SCHEMA

-FerrariFastestLapTimes(race, date, year, driver, lap, time)

GAV MAPPING

{ (a, b, c, d, e, f) | results(y1, x1, x2, x3, y2, …, e, …, f, …),
races(x1, c, y3, y4, a, b, y5, y6),
drivers(x2, y7, y8, y9, y10, d, y11, y12, y13),
constructors(x3, y14, “Ferrari”, y15, y16),
f != “\N”,
c >= 2014 }
-> FerrariFastestLapTimes(a, b, c, d, e, f)

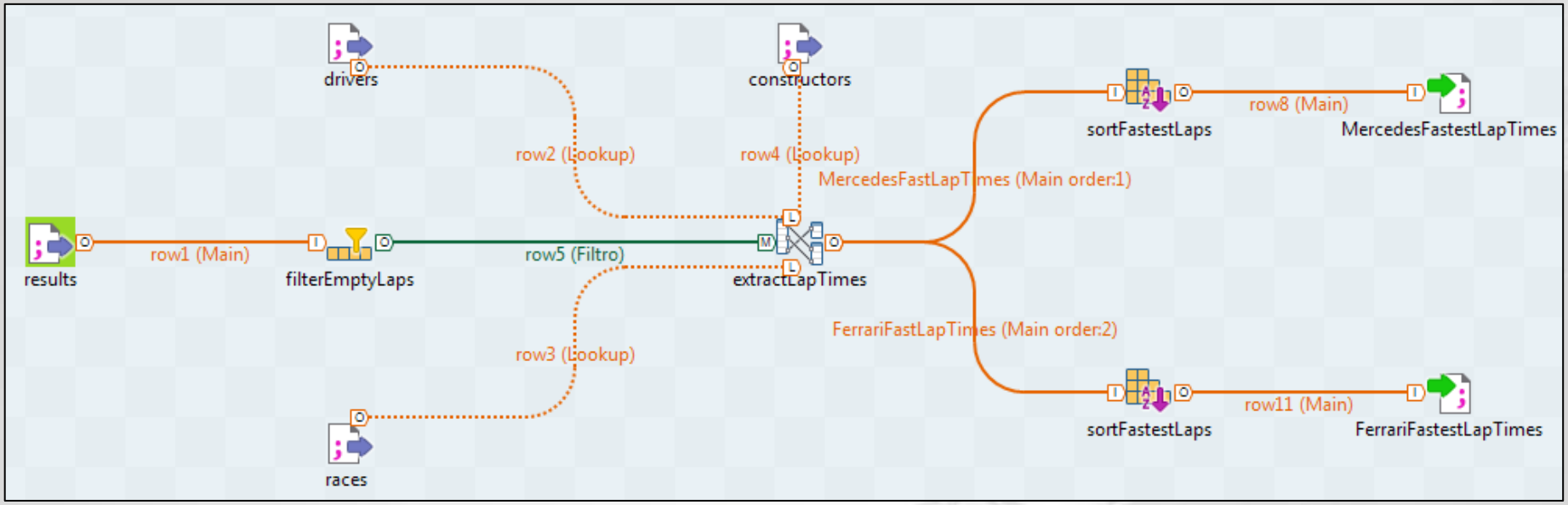
“\N” doesn’t mean that the value of a fastest lap is unknown, but that the driver didn’t have the chance to register one!

GLOBAL SCHEMA

-MercedesFastestLapTimes(race, date, year, driver, lap, time)

GAV MAPPING

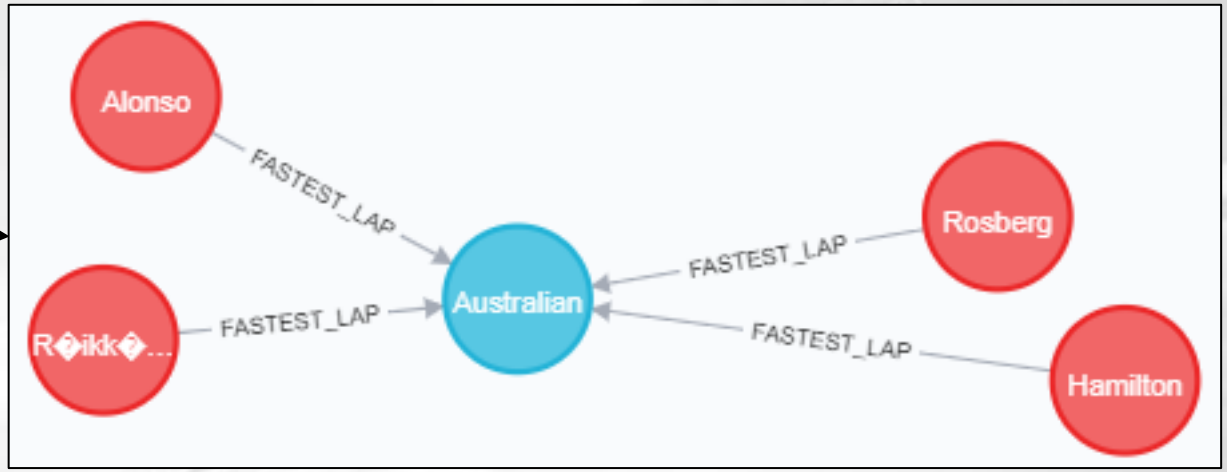
```
{ (a, b, c, d, e, f) | results(y1, x1, x2, x3, y2, …, e, …, f, …),
                       races(x1, c, y3, y4, a, b, y5, y6),
                       drivers(x2, y7, y8, y9, y10, d, y11, y12, y13),
                       constructors(x3, y14, “Mercedes”, y15, y16),
                       f != “\N”,
                       c >= 2014 }
                       -> MercedesFastestLapTimes(a, b, c, d, e, f)
```



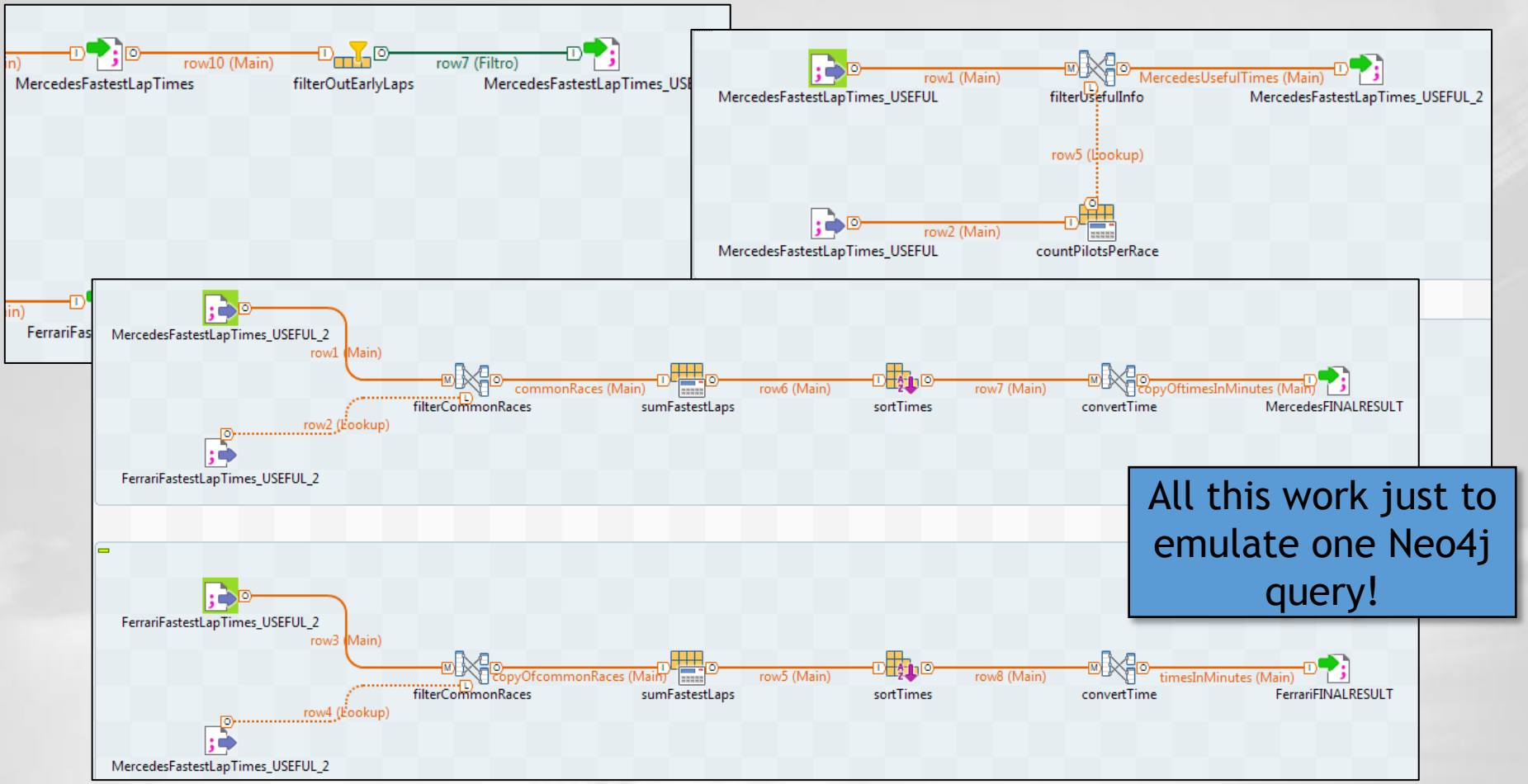
- Now let's see how this works in practice with a live demonstration using Talend!

- Once again we can query the integrated data after it has been loaded in a graph database.

race	date	year	surname	lap	time
Australian Grand Prix	########	2014	Alonso	57	01:33.186
Australian Grand Prix	########	2014	Räikköner	56	01:33.210



- But let's keep using Talend!



- Now let's see how this works in practice with a live demonstration using Talend!

CONCLUSIONS

year	surname	total_time
2014	Alonso	23:03.728
2014	Räikköner	23:08.198
2015	Räikköner	26:46.978
2015	Vettel	26:39.362
2016	Vettel	25:57.157
2016	Räikköner	26:33.142
2017	Räikköner	21:52.122
2017	Vettel	21:47.379
2018	Vettel	27:26.812
2018	Räikköner	27:33.768
2019	Leclerc	28:07.056
2019	Vettel	27:58.008

year	surname	total_time
2014	Rosberg	22:51.041
2014	Hamilton	22:53.061
2015	Rosberg	26:30.006
2015	Hamilton	26:30.009
2016	Rosberg	25:59.515
2016	Hamilton	25:55.375
2017	Bottas	21:50.936
2017	Hamilton	21:46.959
2018	Bottas	27:29.129
2018	Hamilton	27:25.946
2019	Bottas	27:48.603
2019	Hamilton	27:48.313

- This data shows that we clearly need to keep improving the performance of our car because Mercedes is still ahead!

FINAL REMARKS

- By integrating and querying data, I could take a authentic look at how various aspects of the team have performed and where - and if - they fell short.

- The answers I gained from this analysis will help me make decisions in the future that will be based on facts and not subjective feelings, which was exactly my initial objective!

BONUS

Let’s make more mappings!

SOURCE SCHEMAS

- results(resultId, raceId, driverId, constructorId, number, …, position, …, statusId)
- drivers(driverId, driverRef, number, code, forename, surname, dob, nationality, url)
- races(raceId, year, round, circuitId, name, date, time, url)
- constructors(constructorId, constructorRef, name, nationality, url)
- circuits(circuitId, circuitRef, name, location, country, lat, alt, url)
- status(statusId, status)

GLOBAL SCHEMAS

- FerrariWins(race, date, year, surname, name)
- FerrariDNF(race, date, year, surname, name, status)
- italianCircuits(name, location)
- italianDrivers(surname, name, dob)
- italianConstructors(name)

GLOBAL SCHEMAS

- FerrariWins(race, date, year, surname, name)
- FerrariDNF(race, date, year, surname, name, status)

GAV MAPPINGS

```
{ (a, b, c, d, e) | results(y1, x1, x2, x3, y2, …, 1, …),
                    races(x1, c, y3, y4, a, b, y5, y6),
                    drivers(x2, y7, y8, y9, d, e, y10, y11, y12),
                    constructors(x3, y13, “Ferrari”, y14, y15) }
                    -> FerrariWins(a, b, c, d, e)
{ (a, b, c, d, e, f) | results(y1, x1, x2, x3, y2, …, “\N”, …, x4),
                       races(x1, c, y3, y4, a, b, y5, y6),
                       drivers(x2, y7, y8, y9, d, e, y10, y11, y12),
                       constructors(x3, y13, “Ferrari”, y14, y15)
                       status(x4, f) }
                       -> FerrariDNF (a, b, c, d, e, f)
```

GLOBAL SCHEMAS

- italianCircuits(name, location)
- italianDrivers(surname, name, dob)
- italianConstructors(name)

GAV MAPPINGS

{ (a, b) | circuits(y1, y2, a, b, “Italy”, y3, y4, y5) }
-> italianCircuits(a, b)
{ (a, b, c) | drivers(y1, y2, y3, y4, b, a, c, “Italian”, y5) }
-> italianDrivers(a, b, c)
{ (a) | constructors(y1, y2, a, “Italian”, y3) }
-> ItalianConstructors(a)

SOURCE SCHEMAS

- pitStops(raceId, driverId, stop, lap, time, duration, milliseconds)
- results(resultId, raceId, driverId, constructorId, number, grid, position, …, fastestLap, …, fastestLapTime, …, statusId)
- drivers(driverId, driverRef, number, code, forename, surname, dob, nationality, url)
- races(raceId, year, round, circuitId, name, date, time, url)
- constructors(constructorId, constructorRef, name, nationality, url)
- circuits(circuitId, circuitRef, name, location, country, lat, alt, url)
- status(statusId, status)

GLOBAL SCHEMAS

- Pitstops(race, circuit, date, year, surname, name, stop, lap, duration, constructor)
- Results(race, circuit, date, year, surname, name, constructor, grid, position, status)
- FastestLaps(race, circuit, date, year, surname, name, constructor, fastestLap, fastestLapTime)

GLOBAL SCHEMA

- Pitstops(race, circuit, date, year, surname, name, stop, lap, duration, constructor)

GAV MAPPING

{ (a, b, c, d, e, f, g, h, i, l) |
pitStops(x1, x2, g, h, y1, i, y2),
races(x1, d, y3, x4, a, c, y4, y5),
drivers(x2, y6, y7, y8, f, e, y9, y10, y11),
results(y12, x1, x2, x3, y13, …)
constructors(x3, y14, l, y15, y16)
circuits(x4, y17, b, y18, y19, y20, y21, y22) }
-> Pitstops(a, b, c, d, e, f, g, h, i, l)

GLOBAL SCHEMA

- Results(race, circuit, date, year, surname, name, constructor, grid, position, status)

GAV MAPPING

{ (a, b, c, d, e, f, g, h, i, l) |
results(y1, x1, x2, x3, y2, h, i, …, x5)
races(x1, d, y3, x4, a, c, y4, y5),
drivers(x2, y6, y7, y8, f, e, y9, y10, y11),
constructors(x3, y14, g, y15, y16)
circuits(x4, y17, b, y18, y19, y20, y21, y22)
status(x5, l) }
-> Results(a, b, c, d, e, f, g, h, i, l)

GLOBAL SCHEMA

- FastestLaps(race, circuit, date, year, surname, name, constructor, fastestLap, fastestLapTime)

GAV MAPPING

{ (a, b, c, d, e, f, g, h, i) |
results(y1, x1, x2, x3, y2, …, h …, i, …)
races(x1, d, y3, x4, a, c, y4, y5),
drivers(x2, y6, y7, y8, f, e, y9, y10, y11),
constructors(x3, y14, g, y15, y16)
circuits(x4, y17, b, y18, y19, y20, y21, y22)
h != “\N” }
-> FastestLaps(a, b, c, d, e, f, g, h, i)

REFERENCES

- Kaggle F1 Dataset: https://www.kaggle.com/rohanrao/formula-1-world-championship-1950-2020

- Talend Website: https://www.talend.com/

- Neo4j Website: https://neo4j.com/