Università
degli Studi
di Padova

Dipartimento
di Ingegneria
dell'Informazione

TM

Christian Marchiori
Fabio Zanini

# Dataset

# Dataset

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

The dataset consists of all information on the Formula 1 races, drivers, constructors, qualifying, circuits, lap times, pit stops, championships from 1950 till the latest 2023 season.

https://ratings-api.ea.com/v2/entities/f1-23-drivers-ratings

https://github.com/toUpperCase78/formula1-datasets

Driver Ratings from EA & Codemasters F1 2021 / F1 22 / F1 23 Official Video Game

https://github.com/Imagin-io/country-nationality-list/tree/master

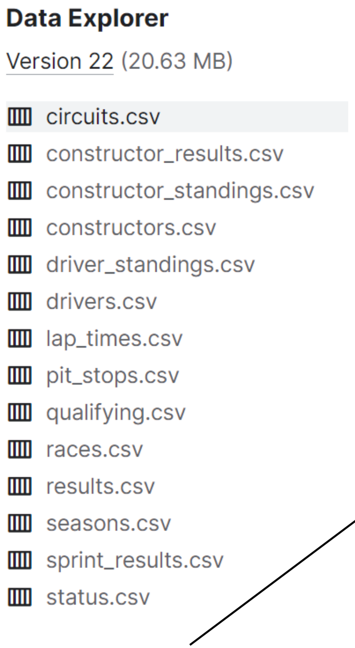
csv file that maps countries to nationalities.

Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

# Dataset

- **circuits.csv** (circuits where F1 races are held)
- **constructor_results.csv** (race results of the constructor's championship)
- **constructor_standings.csv** (final standings of the constructor's championship)
- **constructors.csv** (constructors in F1)
- **driver_standings.csv** (final standings of the driver's championship)
- **drivers.csv** (drivers in F1)
- **lap_times.csv** (lap times in F1)
- **pit_stops.csv** (pit stops in F1)
- **qualifying.csv** (qualifying in F1)
- **races.csv** (races in F1)
- **results.csv** (results of F1 races)
- **seasons.csv** (seasons of F1)
- **sprint_results.csv** (results of F1 sprint races)
- **status.csv** (Mapping of various statuses)

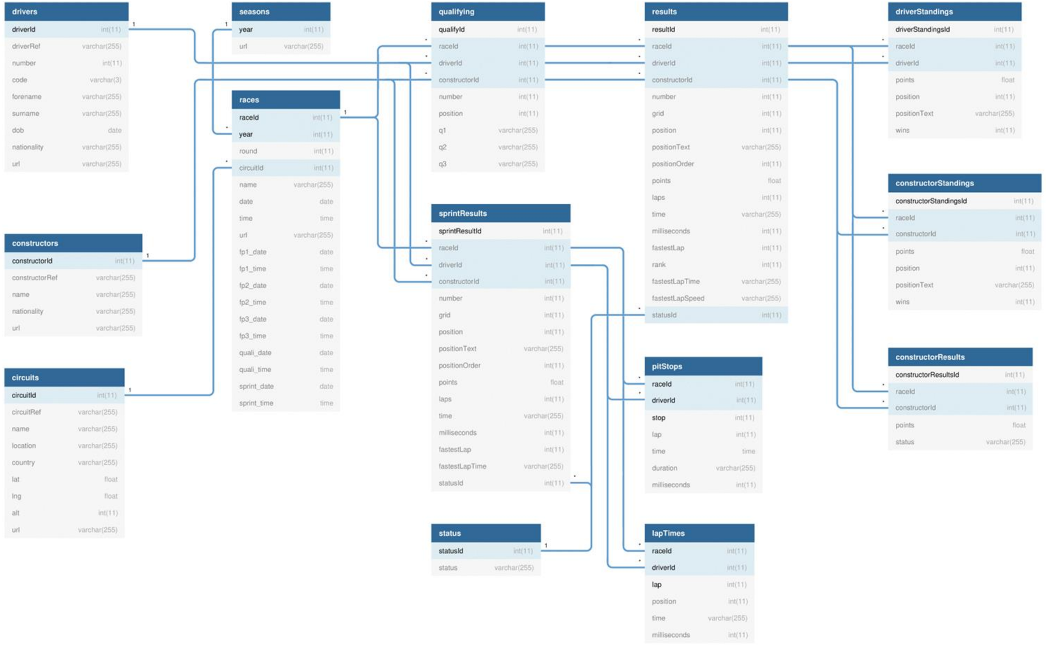


**Data Explorer**

Version 22 (20.63 MB)

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

# E-R Schema



| drivers | |
|---|---|
| driverId | int(11) |
| driverRef | varchar(255) |
| number | int(11) |
| code | varchar(3) |
| forename | varchar(255) |
| surname | varchar(255) |
| dob | date |
| nationality | varchar(255) |
| url | varchar(255) |

| seasons | |
|---|---|
| year | int(11) |
| url | varchar(255) |

| races | |
|---|---|
| raceId | int(11) |
| year | int(11) |
| round | int(11) |
| circuitId | int(11) |
| name | varchar(255) |
| date | date |
| time | time |
| url | varchar(255) |
| fp1_date | date |
| fp1_time | time |
| fp2_date | date |
| fp2_time | time |
| fp3_date | date |
| fp3_time | time |
| quali_date | date |
| quali_time | time |
| sprint_date | date |
| sprint_time | time |

| constructors | |
|---|---|
| constructorId | int(11) |
| constructorRef | varchar(255) |
| name | varchar(255) |
| nationality | varchar(255) |
| url | varchar(255) |

| circuits | |
|---|---|
| circuitId | int(11) |
| circuitRef | varchar(255) |
| name | varchar(255) |
| location | varchar(255) |
| country | varchar(255) |
| lat | float |
| lng | float |
| alt | int(11) |
| url | varchar(255) |

| qualifying | |
|---|---|
| qualifyId | int(11) |
| raceId | int(11) |
| driverId | int(11) |
| constructorId | int(11) |
| number | int(11) |
| position | int(11) |
| q1 | varchar(255) |
| q2 | varchar(255) |
| q3 | varchar(255) |

| sprintResults | |
|---|---|
| sprintResultId | int(11) |
| raceId | int(11) |
| driverId | int(11) |
| constructorId | int(11) |
| number | int(11) |
| grid | int(11) |
| position | int(11) |
| positionText | varchar(255) |
| positionOrder | int(11) |
| points | float |
| laps | int(11) |
| time | varchar(255) |
| milliseconds | int(11) |
| fastestLap | int(11) |
| fastestLapTime | varchar(255) |
| statusId | int(11) |

| status | |
|---|---|
| statusId | int(11) |
| status | varchar(255) |

| results | |
|---|---|
| resultId | int(11) |
| raceId | int(11) |
| driverId | int(11) |
| constructorId | int(11) |
| number | int(11) |
| grid | int(11) |
| position | int(11) |
| positionText | varchar(255) |
| positionOrder | int(11) |
| points | float |
| laps | int(11) |
| time | varchar(255) |
| milliseconds | int(11) |
| fastestLap | int(11) |
| rank | int(11) |
| fastestLapTime | varchar(255) |
| fastestLapSpeed | varchar(255) |
| statusId | int(11) |

| pitStops | |
|---|---|
| raceId | int(11) |
| driverId | int(11) |
| stop | int(11) |
| lap | int(11) |
| time | time |
| duration | varchar(255) |
| milliseconds | int(11) |

| lapTimes | |
|---|---|
| raceId | int(11) |
| driverId | int(11) |
| lap | int(11) |
| position | int(11) |
| time | varchar(255) |
| milliseconds | int(11) |

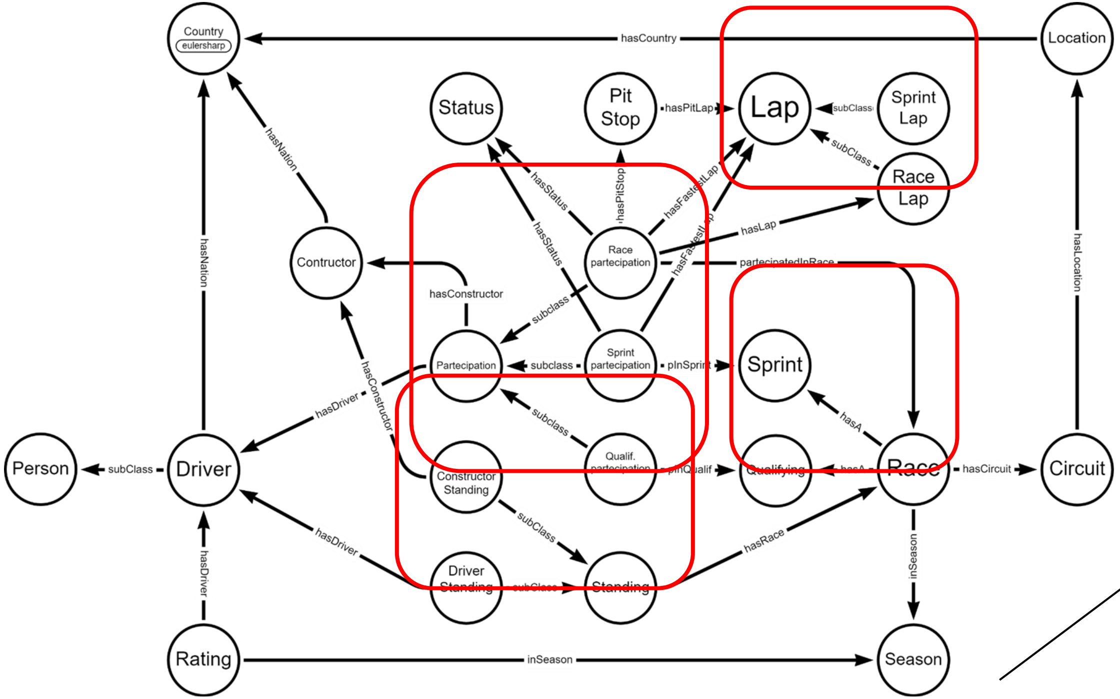
| driverStandings | |
|---|---|
| driverStandingsId | int(11) |
| raceId | int(11) |
| driverId | int(11) |
| points | float |
| position | int(11) |
| positionText | varchar(255) |
| wins | int(11) |

| constructorStandings | |
|---|---|
| constructorStandingsId | int(11) |
| raceId | int(11) |
| constructorId | int(11) |
| points | float |
| position | int(11) |
| positionText | varchar(255) |
| wins | int(11) |

| constructorResults | |
|---|---|
| constructorResultsId | int(11) |
| raceId | int(11) |
| constructorId | int(11) |
| points | float |
| status | varchar(255) |

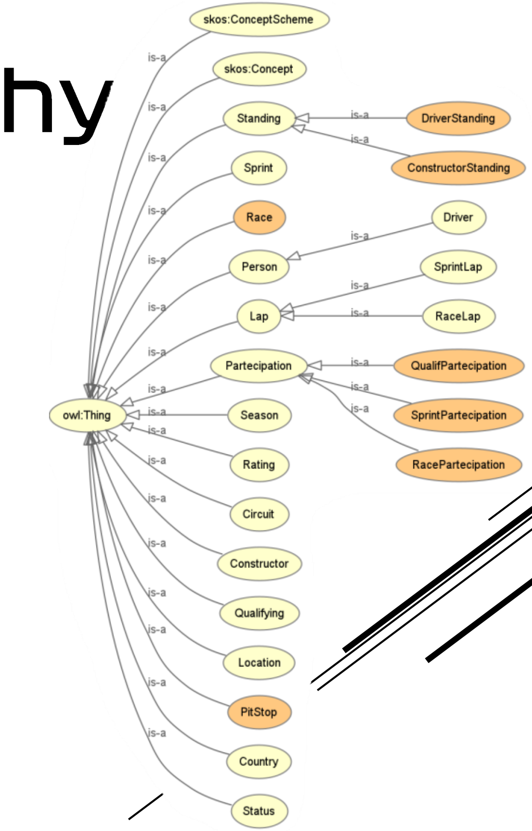
# Ontology

# Graph Schema



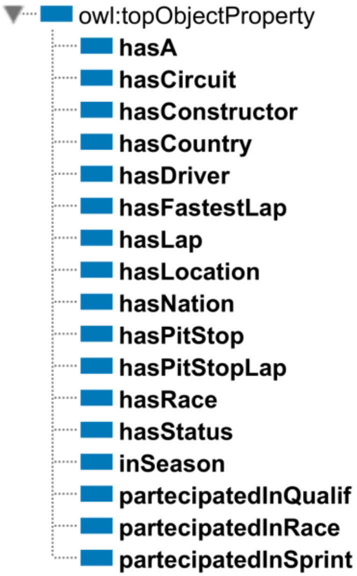
# Ontology - Class hierarchy

- **Standing** (represents championship standings)
  - **DriverStanding** (...related to drivers)
  - **ConstructorStanding** (...related to constructors)
- **Sprint** (represents a sprint race related to a particular race)
- **Race** (represents a particular F1 race)
- **Person** (represents a person in general)
  - **Driver** (represents a F1 driver)
- **Lap** (represents a single lap in a generic race)
  - **SprintLap** (represents a single lap in a sprint)
  - **RaceLap** ([represents a single lap in a race)
- **Partecipation** (represents the participation of a driver and his related constructor team to a race/sprint/qualifying
  - **QualifPartecipation** (...to a qualifying)
  - **SprintPartecipation** (...to a sprint)
  - **RacePartecipation** (...to a race)
- **Season** (represents a F1 season)
- **Rating** (represents a rating tab for a driver)
- **Circuit** (represents the circuits on which F1 races are done)
- **Constructor** (represents F1 teams)
- **Qualifying** (represents a qualifying related to a particular race)
- **Location** (represents the location/city of a circuit)
- **PitStop** (represents a single pitstop in a F1 race)
- **Country** (represents world countries)
- **Status** (represents a status of a driver or a car in a race/sprint)



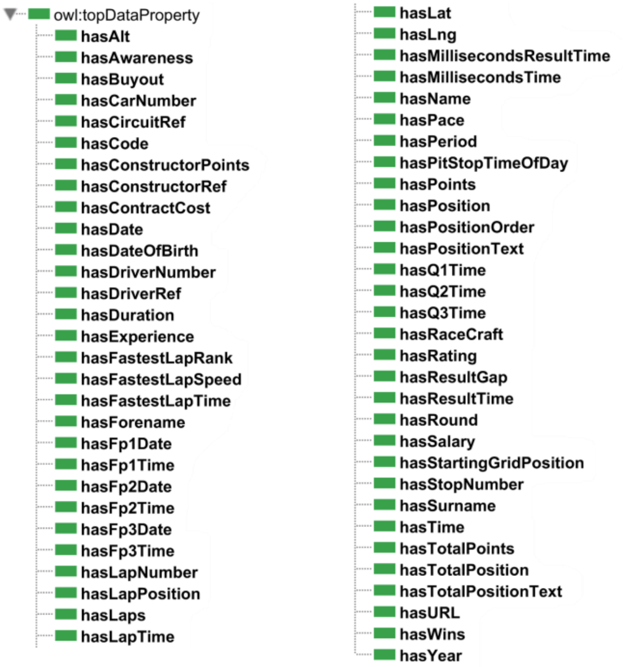
# Ontology - Object properties

- **hasA** (Race ↦ Qualifying or Sprint)
- **hasCircuit** (Race ↦ Circuit)
- **hasConstructor** (ConstructorStanding or Partecipation ↦ Constructor)
- **hasCountry** (Location ↦ Country)
- **hasDriver** (DriverStanding or Partecipation or Rating ↦ Driver)
- **hasFastestLap** (RacePartecipation or SprintPartecipation ↦ Lap)
- **hasLap** (RacePartecipation ↦ Lap)
- **hasLocation** (Circuit ↦ Location)
- **hasNation** (Constructor or Driver ↦ Country)
- **hasPitStop** (RacePartecipation ↦ PitStop)
- **hasPitStopLap** (PitStop ↦ Lap)
- **hasRace** (Standing ↦ Race)
- **hasStatus** (RacePartecipation or SprintPartecipation ↦ Status)
- **inSeason** (Race or Rating ↦ Season)
- **partecipatedInQualif** (QualifPartecipation ↦ Qualifying)
- **partecipatedInRace** (RacePartecipation ↦ Race)
- **partecipatedInSprint** (SprintPartecipation ↦ Sprint)



owl:topObjectProperty
- hasA
- hasCircuit
- hasConstructor
- hasCountry
- hasDriver
- hasFastestLap
- hasLap
- hasLocation
- hasNation
- hasPitStop
- hasPitStopLap
- hasRace
- hasStatus
- inSeason
- partecipatedInQualif
- partecipatedInRace
- partecipatedInSprint

# Ontology - Data properties



- owl:topDataProperty
  - hasAlt
  - hasAwareness
  - hasBuyout
  - hasCarNumber
  - hasCircuitRef
  - hasCode
  - hasConstructorPoints
  - hasConstructorRef
  - hasContractCost
  - hasDate
  - hasDateOfBirth
  - hasDriverNumber
  - hasDriverRef
  - hasDuration
  - hasExperience
  - hasFastestLapRank
  - hasFastestLapSpeed
  - hasFastestLapTime
  - hasForename
  - hasFp1Date
  - hasFp1Time
  - hasFp2Date
  - hasFp2Time
  - hasFp3Date
  - hasFp3Time
  - hasLapNumber
  - hasLapPosition
  - hasLaps
  - hasLapTime
  - hasLat
  - hasLng
  - hasMillisecondsResultTime
  - hasMillisecondsTime
  - hasName
  - hasPace
  - hasPeriod
  - hasPitStopTimeOfDay
  - hasPoints
  - hasPosition
  - hasPositionOrder
  - hasPositionText
  - hasQ1Time
  - hasQ2Time
  - hasQ3Time
  - hasRaceCraft
  - hasRating
  - hasResultGap
  - hasResultTime
  - hasRound
  - hasSalary
  - hasStartingGridPosition
  - hasStopNumber
  - hasSurname
  - hasTime
  - hasTotalPoints
  - hasTotalPosition
  - hasTotalPositionText
  - hasURL
  - hasWins
  - hasYear

# Ontology - Data properties

### DRIVER:

| Property | Datatype |
|---|---|
| hasCode | xsd:integer |
| hasDateOfBirth | xsd:string |
| hasDriverNumber | xsd:integer |
| hasDriverRef | xsd:string |
| hasForename | xsd:string |
| hasSurname | xsd:string |
| hasURL | xsd:string |

### SEASON:

| Property | Datatype |
|---|---|
| hasURL | xsd:string |
| hasYear | xsd:gYear |

### QUALIFYING PARTECIPATION:

| Property | Datatype |
|---|---|
| hasQ1Time | xsd:time |
| hasQ2Time | xsd:time |
| hasQ3Time | xsd:time |

### CIRCUIT:

| Property | Datatype |
|---|---|
| hasAlt | xsd:float |
| hasCircuitRef | xsd:string |
| hasLat | xsd:float |
| hasLng | xsd:float |
| hasName | xsd:string |
| hasUrl | xsd:string |

### PARTECIPATION:

| Property | Datatype |
|---|---|
| hasCarNumber | xsd:integer |
| hasPosition | xsd:integer |

### QUALIFYING and SPRINT:

| Property | Datatype |
|---|---|
| hasDate | xsd:date |
| hasTime | xsd:time |

### CONSTRUCTOR:

| Property | Datatype |
|---|---|
| hasConstructorRef | xsd:string |
| hasName | xsd:string |
| hasUrl | xsd:string |

### RACE PARTECIPATION:

| Property | Datatype |
|---|---|
| hasConstructorPoints | xsd:integer |
| hasFastestLapRank | xsd:integer |
| hasFastestLapSpeed | xsd:decimal |
| hasFastestLapTime | xsd:time |
| hasLaps | xsd:integer |
| hasMillisecondsResultTime | xsd:integer |
| hasPoints | xsd:integer |
| hasPositionOrder | xsd:integer |
| hasPositionText | xsd:string |
| hasResultGap | xsd:time |
| hasResultTime | xsd:time |
| hasStartingGridPosition | xsd:integer |

### PIT STOP:

| Property | Datatype |
|---|---|
| hasDuration | xsd:time |
| hasMillisecondsTime | xsd:integer |
| hasPitStopTimeOfDay | xsd:time |
| hasStopNumber | xsd:integer |

### STANDING:

| Property | Datatype |
|---|---|
| hasTotalPoints | xsd:integer |
| hasTotalPositionOrder | xsd:integer |
| hasTotalPositionText | xsd:string |
| hasWins | xsd:integer |

### SPRINT PARTECIPATION:

| Property | Datatype |
|---|---|
| hasFastestLap | xsd:integer |
| hasFastestLapTime | xsd:time |
| hasLaps | xsd:integer |
| hasMillisecondsResultTime | xsd:integer |
| hasPoints | xsd:integer |
| hasPositionOrder | xsd:integer |
| hasPositionText | xsd:string |
| hasResultGap | xsd:time |
| hasResultTime | xsd:time |
| hasStartingGridPosition | xsd:integer |

### LAP:

| Property | Datatype |
|---|---|
| hasLapNumber | xsd:integer |
| hasLapTime | xsd:time |

### RACE:

| Property | Datatype |
|---|---|
| hasDate | xsd:date |
| hasTime | xsd:time |
| hasFp1Date | xsd:date |
| hasFp2Date | xsd:date |
| hasFp3Date | xsd:date |
| hasFp1Time | xsd:time |
| hasFp2Time | xsd:time |
| hasFp3Time | xsd:time |
| hasName | xsd:string |
| hasRound | xsd:integer |
| hasURL | xsd:string |

### RACE LAP:

| Property | Datatype |
|---|---|
| hasLapPosition | xsd:integer |
| hasMillisecondsTime | xsd:integer |

### RATING:

| Property | Datatype |
|---|---|
| hasAwarness | xsd:integer |
| hasBuyout | xsd:long |
| hasContractCost | xsd:long |
| hasExperience | xsd:integer |
| hasPace | xsd:integer |
| hasPeriod | xsd:date |
| hasRaceCraft | xsd:integer |
| hasRating | xsd:integer |
| hasSalary | xsd:long |

### STATUS:

| Property | Datatype |
|---|---|
| hasName | xsd:string |

# Serialization

Serialization

# Introduction

## Turtle files produced by the serialization process:

| | |
|---|---|
| circuits.ttl | circuits.csv, countries.csv |
| constructors.ttl | constructors.csv, nationalities.csv |
| drivers.ttl | drivers.csv, nationalities.csv |
| laps1.ttl and laps2.ttl | lap_times.csv, results.csv, constructor_results.csv |
| qualifying.ttl | qualifying.csv, races.csv |
| race.ttl | races.csv |
| race_partecipation.ttl | results.csv, constructor_results.csv, lap_times.csv |
| ratings.ttl | ratings.csv, drivers.csv |
| sprint.ttl | sprint_results.csv, races.csv |
| standings.ttl | driver_standings.csv, constructor_standings.csv |
| status_season.ttl | status.csv, seasons.csv |
| stops.ttl | pit_stops.csv, results.csv, constructor_results.csv, lap_times.csv |

Serialization

# Matching columns

1. Left Join

   We joined *results.csv* and *constructor_results.csv* in a single pandas dataframe. Then all the results of constructors (points) are matched with the driver result before managing the data frame.

1. Search

   We searched the row in the original table by matching some constraints.

Serialization

# Nationality to country mapping

We used this approach in drivers.ttl and constructors.ttl to maintain a single class country taken by an external source.

In this way, we avoided creating an additional nationality class.

Thus, it was not necessary to link each individual in the nationality class to the country class via an object property.

The mapping was done directly in the serialization step.

# Queries

## Final 2022 season driver's classification

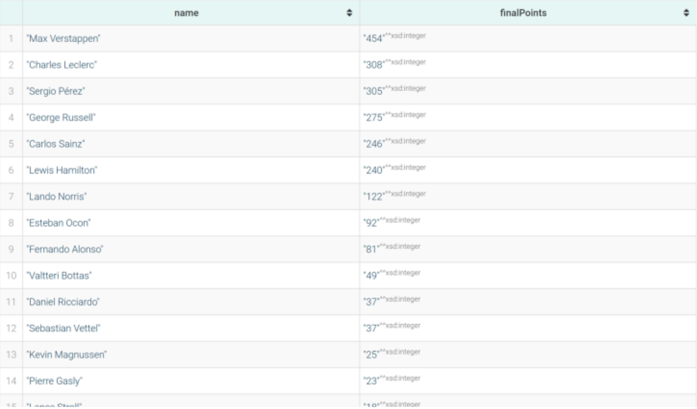
```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?name (MAX(?totPoints) AS ?finalPoints) WHERE {
    ?season :hasYear "2022"^^xsd:integer;
            a :Season.
    ?race :inSeason ?season;
          a :Race.
    ?drivStanding a :DriverStanding;
                  :hasTotalPoints ?totPoints;
                  :hasDriver ?driver;
                  :hasRace ?race.
    ?driver a :Driver;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
}
GROUP BY (?name)
ORDER BY DESC (?finalPoints)
```

## Final 2022 season constructor's classification

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?name (MAX(?totPoints) AS ?finalPoints) WHERE {
    ?season :hasYear "2022"^^xsd:integer;
            a :Season.
    ?race :inSeason ?season;
          a :Race.
    ?consStanding a :ConstructorStanding;
                  :hasTotalPoints ?totPoints;
                  :hasConstructor ?constructor;
                  :hasRace ?race.
    ?constructor :hasName ?name;
                 a :Constructor.
}
GROUP BY (?name)
ORDER BY DESC (?finalPoints)
```

## Final 2022 season driver's classification

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?name (MAX(?totPoints) AS ?finalPoints) WHERE {
    ?season :hasYear "2022"^^xsd:integer;
            a :Season.
    ?race :inSeason ?season;
          a :Race.
    ?drivStanding a :DriverStanding;
                  :hasTotalPoints ?totPoints;
                  :hasDriver ?driver;
                  :hasRace ?race.
    ?driver a :Driver;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
}
GROUP BY (?name)
ORDER BY DESC (?finalPoints)
```



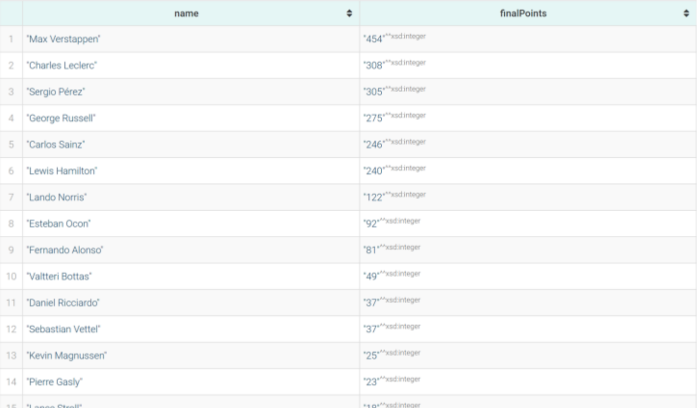
| | name | finalPoints |
|---|---|---|
| 1 | "Max Verstappen" | "454"^^xsd:integer |
| 2 | "Charles Leclerc" | "308"^^xsd:integer |
| 3 | "Sergio Pérez" | "305"^^xsd:integer |
| 4 | "George Russell" | "275"^^xsd:integer |
| 5 | "Carlos Sainz" | "246"^^xsd:integer |
| 6 | "Lewis Hamilton" | "240"^^xsd:integer |
| 7 | "Lando Norris" | "122"^^xsd:integer |
| 8 | "Esteban Ocon" | "92"^^xsd:integer |
| 9 | "Fernando Alonso" | "81"^^xsd:integer |
| 10 | "Valtteri Bottas" | "49"^^xsd:integer |
| 11 | "Daniel Ricciardo" | "37"^^xsd:integer |
| 12 | "Sebastian Vettel" | "37"^^xsd:integer |
| 13 | "Kevin Magnussen" | "25"^^xsd:integer |
| 14 | "Pierre Gasly" | "23"^^xsd:integer |
| 15 | "Lance Stroll" | "18"^^xsd:integer |

## Final 2022 season constructor's classification

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?name (MAX(?totPoints) AS ?finalPoints) WHERE {
    ?season :hasYear "2022"^^xsd:integer;
            a :Season.
    ?race :inSeason ?season;
          a :Race.
    ?consStanding a :ConstructorStanding;
                  :hasTotalPoints ?totPoints;
                  :hasConstructor ?constructor;
                  :hasRace ?race.
    ?constructor :hasName ?name;
                 a :Constructor.
}
GROUP BY (?name)
ORDER BY DESC (?finalPoints)
```

# Final 2022 season driver's classification

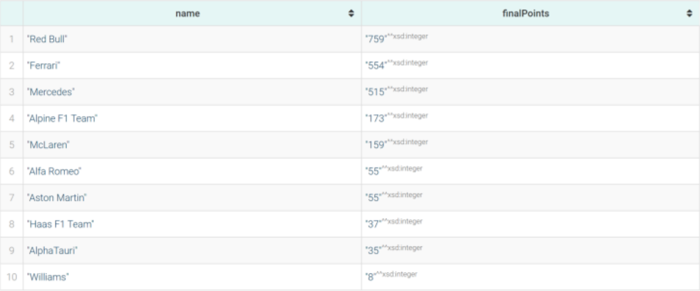
```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?name (MAX(?totPoints) AS ?finalPoints) WHERE {
    ?season :hasYear "2022"^^xsd:integer;
            a :Season.
    ?race :inSeason ?season;
          a :Race.
    ?drivStanding a :DriverStanding;
                  :hasTotalPoints ?totPoints;
                  :hasDriver ?driver;
                  :hasRace ?race.
    ?driver a :Driver;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
}
GROUP BY (?name)
ORDER BY DESC (?finalPoints)
```



| | name | finalPoints |
|---|---|---|
| 1 | "Max Verstappen" | "454"^^xsd:integer |
| 2 | "Charles Leclerc" | "308"^^xsd:integer |
| 3 | "Sergio Pérez" | "305"^^xsd:integer |
| 4 | "George Russell" | "275"^^xsd:integer |
| 5 | "Carlos Sainz" | "246"^^xsd:integer |
| 6 | "Lewis Hamilton" | "240"^^xsd:integer |
| 7 | "Lando Norris" | "122"^^xsd:integer |
| 8 | "Esteban Ocon" | "92"^^xsd:integer |
| 9 | "Fernando Alonso" | "81"^^xsd:integer |
| 10 | "Valtteri Bottas" | "49"^^xsd:integer |
| 11 | "Daniel Ricciardo" | "37"^^xsd:integer |
| 12 | "Sebastian Vettel" | "37"^^xsd:integer |
| 13 | "Kevin Magnussen" | "25"^^xsd:integer |
| 14 | "Pierre Gasly" | "23"^^xsd:integer |
| 15 | "Lance Stroll" | "18"^^xsd:integer |

# Final 2022 season constructor's classification

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?name (MAX(?totPoints) AS ?finalPoints) WHERE {
    ?season :hasYear "2022"^^xsd:integer;
            a :Season.
    ?race :inSeason ?season;
          a :Race.
    ?consStanding a :ConstructorStanding;
                  :hasTotalPoints ?totPoints;
                  :hasConstructor ?constructor;
                  :hasRace ?race.
    ?constructor :hasName ?name;
                 a :Constructor.
}
GROUP BY (?name)
ORDER BY DESC (?finalPoints)
```



| | name | finalPoints |
|---|---|---|
| 1 | "Red Bull" | "759"^^xsd:integer |
| 2 | "Ferrari" | "554"^^xsd:integer |
| 3 | "Mercedes" | "515"^^xsd:integer |
| 4 | "Alpine F1 Team" | "173"^^xsd:integer |
| 5 | "McLaren" | "159"^^xsd:integer |
| 6 | "Alfa Romeo" | "55"^^xsd:integer |
| 7 | "Aston Martin" | "55"^^xsd:integer |
| 8 | "Haas F1 Team" | "37"^^xsd:integer |
| 9 | "AlphaTauri" | "35"^^xsd:integer |
| 10 | "Williams" | "8"^^xsd:integer |

## Complete names of the 20 drivers with the most pit stops made with a single constructor

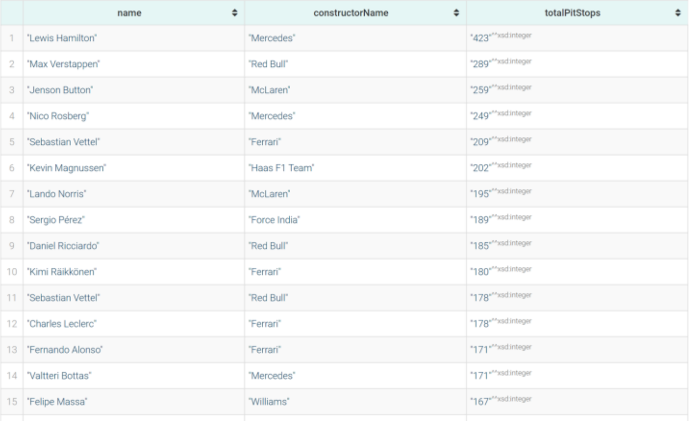
```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?name (?cname AS ?constructorName) (SUM(?maxNumb) AS ?totalPitStops) WHERE{
    {
        SELECT ?partecipation ?name ?cname (MAX(?numb) AS ?maxNumb) WHERE {
            ?driver :hasForename ?fname ;
                    :hasSurname ?sname .
            ?partecipation :hasDriver ?driver ;
                           :hasPitStop ?stop ;
                           :hasConstructor ?constructor .
            ?constructor :hasName ?cname .
            ?stop :hasStopNumber ?numb .
            BIND(CONCAT(?fname, " ", ?sname) AS ?name).
        }
        GROUP BY ?partecipation ?name ?cname
    }
}
GROUP BY ?name ?cname
ORDER BY DESC(?totalPitStops)
LIMIT 20
```

## Most paid driver of 2023 season

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?name (MAX(?sal) as ?salary) WHERE {
    ?season :hasYear "2023"^^xsd:integer;
            a :Season.
    ?rating :inSeason ?season;
            :hasDriver ?driver;
            :hasSalary ?sal;
            a :Rating.
    ?driver a :Driver;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
}
GROUP BY (?name)
ORDER BY DESC (?salary)
```

## Complete names of the 20 drivers with the most pit stops made with a single constructor

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?name (?cname AS ?constructorName) (SUM(?maxNumb) AS ?totalPitStops) WHERE{
    {
        SELECT ?partecipation ?name ?cname (MAX(?numb) AS ?maxNumb) WHERE {
            ?driver :hasForename ?fname ;
                    :hasSurname ?sname .
            ?partecipation :hasDriver ?driver ;
                           :hasPitStop ?stop ;
                           :hasConstructor ?constructor .
            ?constructor :hasName ?cname .
            ?stop :hasStopNumber ?numb .
            BIND(CONCAT(?fname, " ", ?sname) AS ?name).
        }
        GROUP BY ?partecipation ?name ?cname
    }
}
GROUP BY ?name ?cname
ORDER BY DESC(?totalPitStops)
LIMIT 20
```



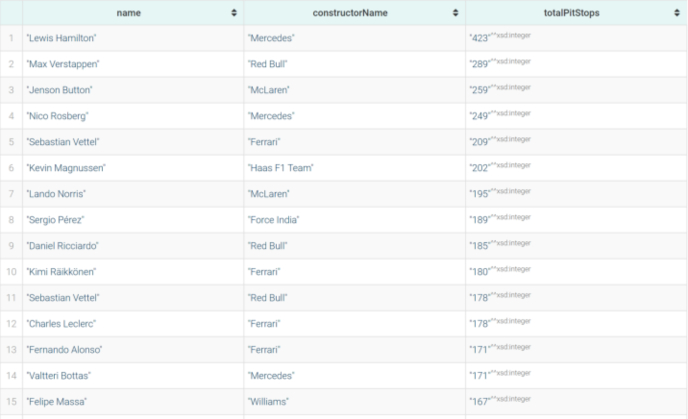
| | name | constructorName | totalPitStops |
|---|---|---|---|
| 1 | "Lewis Hamilton" | "Mercedes" | "423"^^xsd:integer |
| 2 | "Max Verstappen" | "Red Bull" | "289"^^xsd:integer |
| 3 | "Jenson Button" | "McLaren" | "259"^^xsd:integer |
| 4 | "Nico Rosberg" | "Mercedes" | "249"^^xsd:integer |
| 5 | "Sebastian Vettel" | "Ferrari" | "209"^^xsd:integer |
| 6 | "Kevin Magnussen" | "Haas F1 Team" | "202"^^xsd:integer |
| 7 | "Lando Norris" | "McLaren" | "195"^^xsd:integer |
| 8 | "Sergio Pérez" | "Force India" | "189"^^xsd:integer |
| 9 | "Daniel Ricciardo" | "Red Bull" | "185"^^xsd:integer |
| 10 | "Kimi Räikkönen" | "Ferrari" | "180"^^xsd:integer |
| 11 | "Sebastian Vettel" | "Red Bull" | "178"^^xsd:integer |
| 12 | "Charles Leclerc" | "Ferrari" | "178"^^xsd:integer |
| 13 | "Fernando Alonso" | "Ferrari" | "171"^^xsd:integer |
| 14 | "Valtteri Bottas" | "Mercedes" | "171"^^xsd:integer |
| 15 | "Felipe Massa" | "Williams" | "167"^^xsd:integer |

## Most paid driver of 2023 season

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?name (MAX(?sal) as ?salary) WHERE {
    ?season :hasYear "2023"^^xsd:integer;
            a :Season.
    ?rating :inSeason ?season;
            :hasDriver ?driver;
            :hasSalary ?sal;
            a :Rating.
    ?driver a :Driver;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
}
GROUP BY (?name)
ORDER BY DESC (?salary)
```

## Complete names of the 20 drivers with the most pit stops made with a single constructor

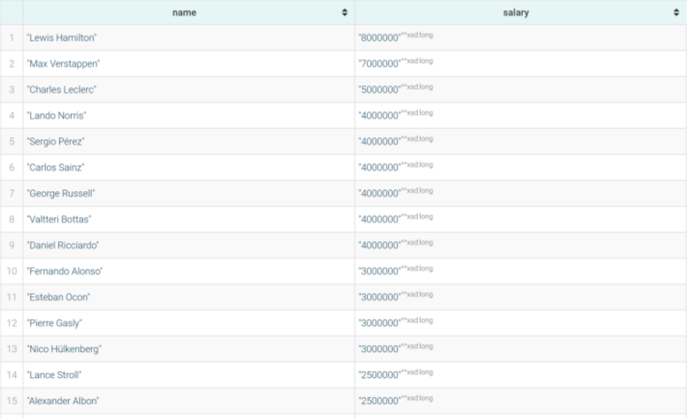
```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?name (?cname AS ?constructorName) (SUM(?maxNumb) AS ?totalPitStops) WHERE{
    {
        SELECT ?partecipation ?name ?cname (MAX(?numb) AS ?maxNumb) WHERE {
            ?driver :hasForename ?fname ;
                    :hasSurname ?sname .
            ?partecipation :hasDriver ?driver ;
                           :hasPitStop ?stop ;
                           :hasConstructor ?constructor .
            ?constructor :hasName ?cname .
            ?stop :hasStopNumber ?numb .
            BIND(CONCAT(?fname, " ", ?sname) AS ?name).
        }
        GROUP BY ?partecipation ?name ?cname
    }
}
GROUP BY ?name ?cname
ORDER BY DESC(?totalPitStops)
LIMIT 20
```



| | name | constructorName | totalPitStops |
|---|---|---|---|
| 1 | "Lewis Hamilton" | "Mercedes" | "423"^^xsd:integer |
| 2 | "Max Verstappen" | "Red Bull" | "289"^^xsd:integer |
| 3 | "Jenson Button" | "McLaren" | "259"^^xsd:integer |
| 4 | "Nico Rosberg" | "Mercedes" | "249"^^xsd:integer |
| 5 | "Sebastian Vettel" | "Ferrari" | "209"^^xsd:integer |
| 6 | "Kevin Magnussen" | "Haas F1 Team" | "202"^^xsd:integer |
| 7 | "Lando Norris" | "McLaren" | "195"^^xsd:integer |
| 8 | "Sergio Pérez" | "Force India" | "189"^^xsd:integer |
| 9 | "Daniel Ricciardo" | "Red Bull" | "185"^^xsd:integer |
| 10 | "Kimi Räikkönen" | "Ferrari" | "180"^^xsd:integer |
| 11 | "Sebastian Vettel" | "Red Bull" | "178"^^xsd:integer |
| 12 | "Charles Leclerc" | "Ferrari" | "178"^^xsd:integer |
| 13 | "Fernando Alonso" | "Ferrari" | "171"^^xsd:integer |
| 14 | "Valtteri Bottas" | "Mercedes" | "171"^^xsd:integer |
| 15 | "Felipe Massa" | "Williams" | "167"^^xsd:integer |

## Most paid driver of 2023 season

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?name (MAX(?sal) as ?salary) WHERE {
    ?season :hasYear "2023"^^xsd:integer;
            a :Season.
    ?rating :inSeason ?season;
            :hasDriver ?driver;
            :hasSalary ?sal;
            a :Rating.
    ?driver a :Driver;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
}
GROUP BY (?name)
ORDER BY DESC (?salary)
```



| | name | salary |
|---|---|---|
| 1 | "Lewis Hamilton" | "8000000"^^xsd:long |
| 2 | "Max Verstappen" | "7000000"^^xsd:long |
| 3 | "Charles Leclerc" | "5000000"^^xsd:long |
| 4 | "Lando Norris" | "4000000"^^xsd:long |
| 5 | "Sergio Pérez" | "4000000"^^xsd:long |
| 6 | "Carlos Sainz" | "4000000"^^xsd:long |
| 7 | "George Russell" | "4000000"^^xsd:long |
| 8 | "Valtteri Bottas" | "4000000"^^xsd:long |
| 9 | "Daniel Ricciardo" | "4000000"^^xsd:long |
| 10 | "Fernando Alonso" | "3000000"^^xsd:long |
| 11 | "Esteban Ocon" | "3000000"^^xsd:long |
| 12 | "Pierre Gasly" | "3000000"^^xsd:long |
| 13 | "Nico Hülkenberg" | "3000000"^^xsd:long |
| 14 | "Lance Stroll" | "2500000"^^xsd:long |
| 15 | "Alexander Albon" | "2500000"^^xsd:long |

## All the italian drivers who won an italian grand prix, driving an italian car, ordered by number of wins

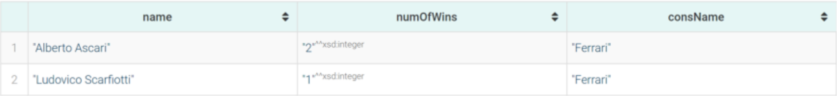
```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX countries: <http://eulersharp.sourceforge.net/2003/03swap/countries#>
SELECT ?name (COUNT(?name) as ?numOfWins) ?consName WHERE {
    ?race :hasCircuit ?circuit;
          a :Race.
    ?circuit :hasLocation ?location;
             a :Circuit.
    ?location :hasCountry countries:it;
              a :Location.
    ?racePart :partecipatedInRace ?race;
              :hasPositionOrder "1"^^xsd:integer;
              :hasConstructor ?constructor;
              :hasDriver ?driver.
    ?constructor a :Constructor;
                 :hasName ?consName;
                 :hasNation countries:it.
    ?driver a :Driver;
            :hasNation countries:it;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
} GROUP BY ?name ?consName
ORDER BY DESC (?numOfWins)
```

## The most winning italian driver in F1 history

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX countries: <http://eulersharp.sourceforge.net/2003/03swap/countries#>
SELECT ?name (COUNT(?name) as ?numOfWins) WHERE {
    ?race :hasCircuit ?circuit;
          a :Race.
    ?racePart :partecipatedInRace ?race;
              :hasPositionOrder "1"^^xsd:integer;
              :hasDriver ?driver.
    ?driver a :Driver;
            :hasNation countries:it;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
} GROUP BY ?name
ORDER BY DESC (?numOfWins)
LIMIT 1
```

## All the italian drivers who won an italian grand prix, driving an italian car, ordered by number of wins

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX countries: <http://eulersharp.sourceforge.net/2003/03swap/countries#>
SELECT ?name (COUNT(?name) as ?numOfWins) ?consName WHERE {
    ?race :hasCircuit ?circuit;
          a :Race.
    ?circuit :hasLocation ?location;
             a :Circuit.
    ?location :hasCountry countries:it;
              a :Location.
    ?racePart :partecipatedInRace ?race;
              :hasPositionOrder "1"^^xsd:integer;
              :hasConstructor ?constructor;
              :hasDriver ?driver.
    ?constructor a :Constructor;
                 :hasName ?consName;
                 :hasNation countries:it.
    ?driver a :Driver;
            :hasNation countries:it;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
} GROUP BY ?name ?consName
ORDER BY DESC (?numOfWins)
```



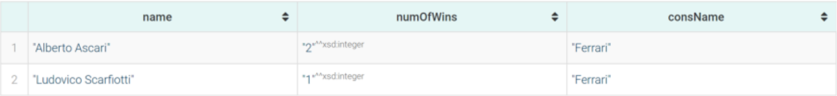
|   | name | numOfWins | consName |
|---|---|---|---|
| 1 | "Alberto Ascari" | "2"^^xsd:integer | "Ferrari" |
| 2 | "Ludovico Scarfiotti" | "1"^^xsd:integer | "Ferrari" |

## The most winning italian driver in F1 history

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX countries: <http://eulersharp.sourceforge.net/2003/03swap/countries#>
SELECT ?name (COUNT(?name) as ?numOfWins) WHERE {
    ?race :hasCircuit ?circuit;
          a :Race.
    ?racePart :partecipatedInRace ?race;
              :hasPositionOrder "1"^^xsd:integer;
              :hasDriver ?driver.
    ?driver a :Driver;
            :hasNation countries:it;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
} GROUP BY ?name
ORDER BY DESC (?numOfWins)
LIMIT 1
```

## All the italian drivers who won an italian grand prix, driving an italian car, ordered by number of wins

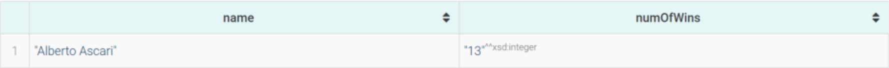
```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX countries: <http://eulersharp.sourceforge.net/2003/03swap/countries#>
SELECT ?name (COUNT(?name) as ?numOfWins) ?consName WHERE {
    ?race :hasCircuit ?circuit;
          a :Race.
    ?circuit :hasLocation ?location;
             a :Circuit.
    ?location :hasCountry countries:it;
              a :Location.
    ?racePart :partecipatedInRace ?race;
              :hasPositionOrder "1"^^xsd:integer;
              :hasConstructor ?constructor;
              :hasDriver ?driver.
    ?constructor a :Constructor;
                 :hasName ?consName;
                 :hasNation countries:it.
    ?driver a :Driver;
            :hasNation countries:it;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
} GROUP BY ?name ?consName
ORDER BY DESC (?numOfWins)
```



| | name | numOfWins | consName |
|---|---|---|---|
| 1 | "Alberto Ascari" | "2"^^xsd:integer | "Ferrari" |
| 2 | "Ludovico Scarfiotti" | "1"^^xsd:integer | "Ferrari" |

## The most winning italian driver in F1 history

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX countries: <http://eulersharp.sourceforge.net/2003/03swap/countries#>
SELECT ?name (COUNT(?name) as ?numOfWins) WHERE {
    ?race :hasCircuit ?circuit;
          a :Race.
    ?racePart :partecipatedInRace ?race;
              :hasPositionOrder "1"^^xsd:integer;
              :hasDriver ?driver.
    ?driver a :Driver;
            :hasNation countries:it;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
} GROUP BY ?name
ORDER BY DESC (?numOfWins)
LIMIT 1
```



| | name | numOfWins |
|---|---|---|
| 1 | "Alberto Ascari" | "13"^^xsd:integer |

## The 5 most winning drivers in F1 history, ordered by number of grand prix wins

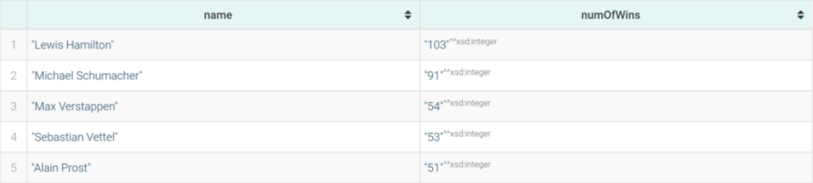
```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?name (COUNT(?name) as ?numOfWins) WHERE {
    ?race :hasCircuit ?circuit;
          a :Race.
    ?racePart :partecipatedInRace ?race;
              :hasPositionOrder "1"^^xsd:integer;
              :hasDriver ?driver.
    ?driver a :Driver;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
} GROUP BY ?name
ORDER BY DESC (?numOfWins)
LIMIT 5
```

## Information regarding the youngest drivers who have won races, in order of age

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?fullName ?age (COUNT(?pos) AS ?wins) (GROUP_CONCAT(DISTINCT
YEAR(?raceDate) ; separator=",") AS ?year) (GROUP_CONCAT(DISTINCT ?circName ;
separator=",") AS ?circuitNames) WHERE{
    ?part :hasDriver ?driver ;
          :hasPosition ?pos ;
              :partecipatedInRace ?race .
    ?race :hasDate ?raceDate ;
          :hasCircuit ?circ .
    ?circ :hasName ?circName .
    ?driver :hasDateOfBirth ?dob ;
            :hasForename ?fname ;
            :hasSurname ?lname .
    BIND(CONCAT(?fname, " ", ?lname) AS ?fullName)
    BIND((YEAR(?raceDate) - YEAR(?dob)) AS ?age)
    FILTER(?pos = 1 && ?age < 26)
}
GROUP BY ?fullName ?age
ORDER BY ASC (?age)
```

## The 5 most winning drivers in F1 history, ordered by number of grand prix wins

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?name (COUNT(?name) as ?numOfWins) WHERE {
    ?race :hasCircuit ?circuit;
          a :Race.
    ?racePart :partecipatedInRace ?race;
              :hasPositionOrder "1"^^xsd:integer;
              :hasDriver ?driver.
    ?driver a :Driver;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
} GROUP BY ?name
ORDER BY DESC (?numOfWins)
LIMIT 5
```



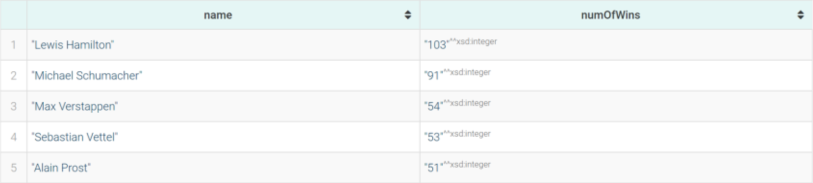
| | name | numOfWins |
|---|---|---|
| 1 | "Lewis Hamilton" | "103"^^xsd:integer |
| 2 | "Michael Schumacher" | "91"^^xsd:integer |
| 3 | "Max Verstappen" | "54"^^xsd:integer |
| 4 | "Sebastian Vettel" | "53"^^xsd:integer |
| 5 | "Alain Prost" | "51"^^xsd:integer |

## Information regarding the youngest drivers who have won races, in order of age

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?fullName ?age (COUNT(?pos) AS ?wins) (GROUP_CONCAT(DISTINCT
YEAR(?raceDate) ; separator=",") AS ?year) (GROUP_CONCAT(DISTINCT ?circName ;
separator=",") AS ?circuitNames) WHERE{
    ?part :hasDriver ?driver ;
          :hasPosition ?pos ;
              :partecipatedInRace ?race .
    ?race :hasDate ?raceDate ;
          :hasCircuit ?circ .
    ?circ :hasName ?circName .
    ?driver :hasDateOfBirth ?dob ;
            :hasForename ?fname ;
            :hasSurname ?lname .
    BIND(CONCAT(?fname, " ", ?lname) AS ?fullName)
    BIND((YEAR(?raceDate) - YEAR(?dob)) AS ?age)
    FILTER(?pos = 1 && ?age < 26)
}
GROUP BY ?fullName ?age
ORDER BY ASC (?age)
```

## The 5 most winning drivers in F1 history, ordered by number of grand prix wins

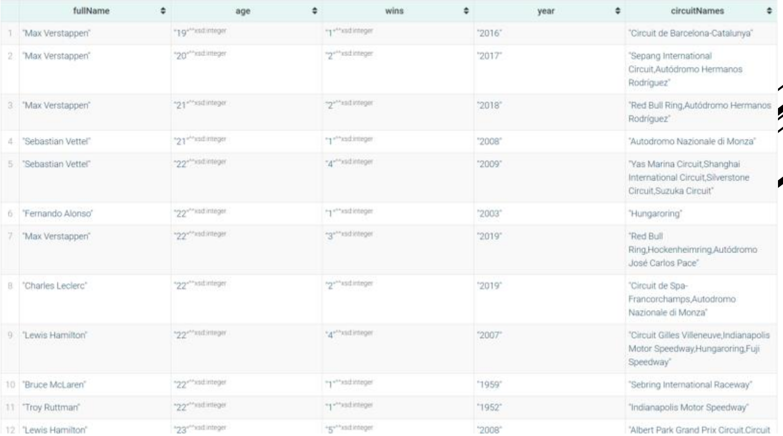
```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?name (COUNT(?name) as ?numOfWins) WHERE {
    ?race :hasCircuit ?circuit;
          a :Race.
    ?racePart :partecipatedInRace ?race;
              :hasPositionOrder "1"^^xsd:integer;
              :hasDriver ?driver.
    ?driver a :Driver;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
} GROUP BY ?name
ORDER BY DESC (?numOfWins)
LIMIT 5
```



|  | name | numOfWins |
|---|---|---|
| 1 | "Lewis Hamilton" | "103"^^xsd:integer |
| 2 | "Michael Schumacher" | "91"^^xsd:integer |
| 3 | "Max Verstappen" | "54"^^xsd:integer |
| 4 | "Sebastian Vettel" | "53"^^xsd:integer |
| 5 | "Alain Prost" | "51"^^xsd:integer |

## Information regarding the youngest drivers who have won races, in order of age

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?fullName ?age (COUNT(?pos) AS ?wins) (GROUP_CONCAT(DISTINCT
YEAR(?raceDate) ; separator=",") AS ?year) (GROUP_CONCAT(DISTINCT ?circName ;
separator=",") AS ?circuitNames) WHERE{
    ?part :hasDriver ?driver ;
          :hasPosition ?pos ;
              :partecipatedInRace ?race .
    ?race :hasDate ?raceDate ;
          :hasCircuit ?circ .
    ?circ :hasName ?circName .
    ?driver :hasDateOfBirth ?dob ;
            :hasForename ?fname ;
            :hasSurname ?lname .
    BIND(CONCAT(?fname, " ", ?lname) AS ?fullName)
    BIND((YEAR(?raceDate) - YEAR(?dob)) AS ?age)
    FILTER(?pos = 1 && ?age < 26)
}
GROUP BY ?fullName ?age
ORDER BY ASC (?age)
```



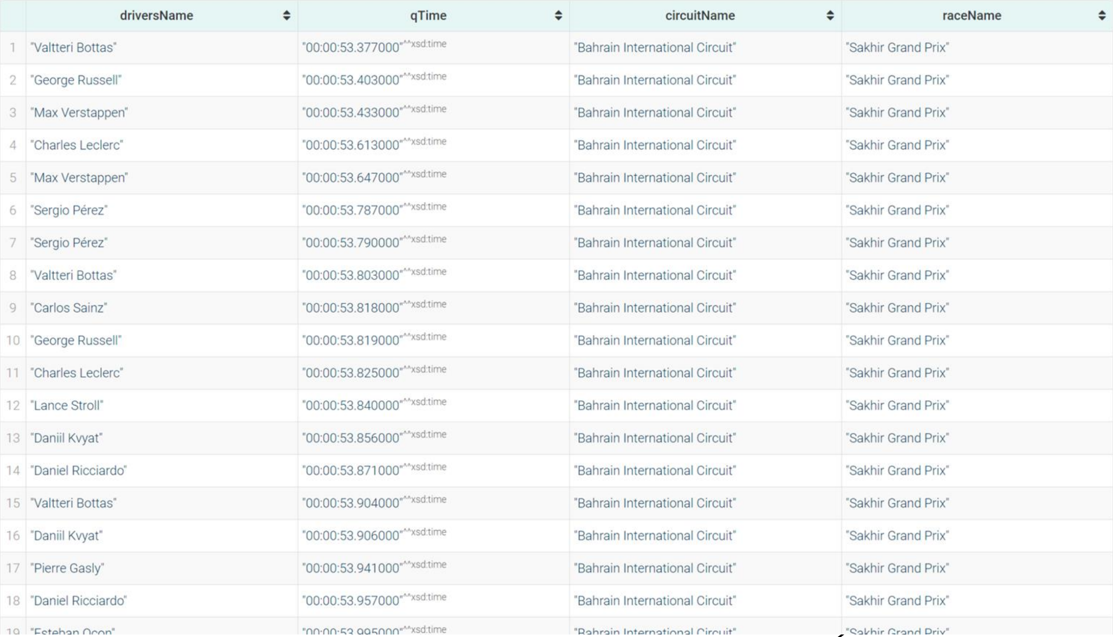
|  | fullName | age | wins | year | circuitNames |
|---|---|---|---|---|---|
| 1 | "Max Verstappen" | "19"^^xsd:integer | "1"^^xsd:integer | "2016" | "Circuit de Barcelona-Catalunya" |
| 2 | "Max Verstappen" | "20"^^xsd:integer | "2"^^xsd:integer | "2017" | "Sepang International Circuit,Autódromo Hermanos Rodríguez" |
| 3 | "Max Verstappen" | "21"^^xsd:integer | "2"^^xsd:integer | "2018" | "Red Bull Ring,Autódromo Hermanos Rodríguez" |
| 4 | "Sebastian Vettel" | "21"^^xsd:integer | "1"^^xsd:integer | "2008" | "Autodromo Nazionale di Monza" |
| 5 | "Sebastian Vettel" | "22"^^xsd:integer | "4"^^xsd:integer | "2009" | "Yas Marina Circuit,Shanghai International Circuit,Silverstone Circuit,Suzuka Circuit" |
| 6 | "Fernando Alonso" | "22"^^xsd:integer | "1"^^xsd:integer | "2003" | "Hungaroring" |
| 7 | "Max Verstappen" | "22"^^xsd:integer | "3"^^xsd:integer | "2019" | "Red Bull Ring,Hockenheimring,Autódromo José Carlos Pace" |
| 8 | "Charles Leclerc" | "22"^^xsd:integer | "2"^^xsd:integer | "2019" | "Circuit de Spa-Francorchamps,Autodromo Nazionale di Monza" |
| 9 | "Lewis Hamilton" | "22"^^xsd:integer | "4"^^xsd:integer | "2007" | "Circuit Gilles Villeneuve,Indianapolis Motor Speedway,Hungaroring,Fuji Speedway" |
| 10 | "Bruce McLaren" | "22"^^xsd:integer | "1"^^xsd:integer | "1959" | "Sebring International Raceway" |
| 11 | "Troy Ruttman" | "22"^^xsd:integer | "1"^^xsd:integer | "1952" | "Indianapolis Motor Speedway" |
| 12 | "Lewis Hamilton" | "23"^^xsd:integer | "5"^^xsd:integer | "2008" | "Albert Park Grand Prix Circuit,Circuit |

# Data about fastest qualifying laps in F1 history (both q1, q2 and q3)

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?driversName ?qTime ?circuitName ?raceName WHERE {
    {
        ?qualPart :hasQ3Time ?qTime.
    }
    UNION
    {
        ?qualPart :hasQ2Time ?qTime.
    }
    UNION
    {
        ?qualPart :hasQ1Time ?qTime.
    }
    ?qualPart :hasDriver ?driver;
              :partecipatedInQualif ?quali;
              a :QualifPartecipation.
    ?quali a :Qualifying.
    ?race a :Race;
          :hasA ?quali;
          :hasName ?raceName;
          :hasCircuit ?circ.
    ?circ a :Circuit;
             :hasName ?circuitName.
    ?driver a :Driver;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?driversName).
    FILTER(?qTime != "00:00:00"^^xsd:time)
}
ORDER BY (?qTime)
LIMIT 100
```

# Data about fastest qualifying laps in F1 history (both q1, q2 and q3)

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?driversName ?qTime ?circuitName ?raceName WHERE {
    {
        ?qualPart :hasQ3Time ?qTime.
    }
    UNION
    {
        ?qualPart :hasQ2Time ?qTime.
    }
    UNION
    {
        ?qualPart :hasQ1Time ?qTime.
    }
    ?qualPart :hasDriver ?driver;
              :partecipatedInQualif ?quali;
              a :QualifPartecipation.
    ?quali a :Qualifying.
    ?race a :Race;
          :hasA ?quali;
          :hasName ?raceName;
          :hasCircuit ?circ.
    ?circ a :Circuit;
             :hasName ?circuitName.
    ?driver a :Driver;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?driversName).
    FILTER(?qTime != "00:00:00"^^xsd:time)
}
ORDER BY (?qTime)
LIMIT 100
```



| | driversName | qTime | circuitName | raceName |
|---|---|---|---|---|
| 1 | "Valtteri Bottas" | "00:00:53.377000"^^xsd:time | "Bahrain International Circuit" | "Sakhir Grand Prix" |
| 2 | "George Russell" | "00:00:53.403000"^^xsd:time | "Bahrain International Circuit" | "Sakhir Grand Prix" |
| 3 | "Max Verstappen" | "00:00:53.433000"^^xsd:time | "Bahrain International Circuit" | "Sakhir Grand Prix" |
| 4 | "Charles Leclerc" | "00:00:53.613000"^^xsd:time | "Bahrain International Circuit" | "Sakhir Grand Prix" |
| 5 | "Max Verstappen" | "00:00:53.647000"^^xsd:time | "Bahrain International Circuit" | "Sakhir Grand Prix" |
| 6 | "Sergio Pérez" | "00:00:53.787000"^^xsd:time | "Bahrain International Circuit" | "Sakhir Grand Prix" |
| 7 | "Sergio Pérez" | "00:00:53.790000"^^xsd:time | "Bahrain International Circuit" | "Sakhir Grand Prix" |
| 8 | "Valtteri Bottas" | "00:00:53.803000"^^xsd:time | "Bahrain International Circuit" | "Sakhir Grand Prix" |
| 9 | "Carlos Sainz" | "00:00:53.818000"^^xsd:time | "Bahrain International Circuit" | "Sakhir Grand Prix" |
| 10 | "George Russell" | "00:00:53.819000"^^xsd:time | "Bahrain International Circuit" | "Sakhir Grand Prix" |
| 11 | "Charles Leclerc" | "00:00:53.825000"^^xsd:time | "Bahrain International Circuit" | "Sakhir Grand Prix" |
| 12 | "Lance Stroll" | "00:00:53.840000"^^xsd:time | "Bahrain International Circuit" | "Sakhir Grand Prix" |
| 13 | "Daniil Kvyat" | "00:00:53.856000"^^xsd:time | "Bahrain International Circuit" | "Sakhir Grand Prix" |
| 14 | "Daniel Ricciardo" | "00:00:53.871000"^^xsd:time | "Bahrain International Circuit" | "Sakhir Grand Prix" |
| 15 | "Valtteri Bottas" | "00:00:53.904000"^^xsd:time | "Bahrain International Circuit" | "Sakhir Grand Prix" |
| 16 | "Daniil Kvyat" | "00:00:53.906000"^^xsd:time | "Bahrain International Circuit" | "Sakhir Grand Prix" |
| 17 | "Pierre Gasly" | "00:00:53.941000"^^xsd:time | "Bahrain International Circuit" | "Sakhir Grand Prix" |
| 18 | "Daniel Ricciardo" | "00:00:53.957000"^^xsd:time | "Bahrain International Circuit" | "Sakhir Grand Prix" |
| 19 | "Esteban Ocon" | "00:00:53.995000"^^xsd:time | "Bahrain International Circuit" | "Sakhir Grand Prix" |

## Name of the F1 driver with the highest number of pole position achieved

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?driversName (COUNT(?driversName) as ?numOfPoles) WHERE {
    ?qualiPart a :QualifPartecipation;
               :hasPosition "1"^^xsd:integer;
               :hasDriver ?driver.
    ?driver a :Driver;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?driversName).
}
GROUP BY (?driversName)
ORDER BY DESC (?numOfPoles)
LIMIT 1
```

## Date of the first qualifying in the dataset

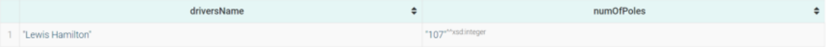
```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?date WHERE {
    ?qualiPart a :QualifPartecipation;
               :partecipatedInQualif ?quali.
    ?quali a :Qualifying.
    ?race a :Race;
          :hasA ?quali;
          :hasDate ?date.
}
ORDER BY (?date)
LIMIT 1
```

## Name of the F1 driver with the highest number of pole position achieved, counted as number of times his starting grid position was 1. Also, printed the qualifying result in those races, to see if qualifying result is pole position or not (for penalties or other stuff)

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?driversName (COUNT(?driversName) as ?numOfPoles)
(GROUP_CONCAT((?qualiPos) ; separator=",") AS ?qualifPositions) WHERE {
    ?racePart a :RacePartecipation;
              :partecipatedInRace ?race;
              :hasStartingGridPosition "1"^^xsd:integer;
              :hasDriver ?driver.
    ?race a :Race.
    ?driver a :Driver;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?driversName).
    OPTIONAL{
        ?race :hasA ?qual.
        ?qual a :Qualifying.
        ?qualiPart a :QualifPartecipation;
               :partecipatedInQualif ?qual;
               :hasDriver ?driver;
               :hasPosition ?qualiPos.
    }.
}
GROUP BY ?driversName
ORDER BY DESC (?numOfPoles)
LIMIT 1
```

## Name of the F1 driver with the highest number of pole position achieved

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?driversName (COUNT(?driversName) as ?numOfPoles) WHERE {
    ?qualiPart a :QualifPartecipation;
               :hasPosition "1"^^xsd:integer;
               :hasDriver ?driver.
    ?driver a :Driver;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?driversName).
}
GROUP BY (?driversName)
ORDER BY DESC (?numOfPoles)
LIMIT 1
```



|  | driversName | numOfPoles |
|---|---|---|
| 1 | "Lewis Hamilton" | "107"^^xsd:integer |

## Date of the first qualifying in the dataset

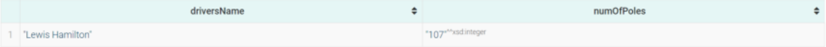
```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?date WHERE {
    ?qualiPart a :QualifPartecipation;
               :partecipatedInQualif ?quali.
    ?quali a :Qualifying.
    ?race a :Race;
          :hasA ?quali;
          :hasDate ?date.
}
ORDER BY (?date)
LIMIT 1
```

## Name of the F1 driver with the highest number of pole position achieved, counted as number of times his starting grid position was 1. Also, printed the qualifying result in those races, to see if qualifying result is pole position or not (for penalties or other stuff)

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?driversName (COUNT(?driversName) as ?numOfPoles)
(GROUP_CONCAT((?qualiPos) ; separator=",") AS ?qualifPositions) WHERE {
    ?racePart a :RacePartecipation;
              :partecipatedInRace ?race;
              :hasStartingGridPosition "1"^^xsd:integer;
              :hasDriver ?driver.
    ?race a :Race.
    ?driver a :Driver;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?driversName).
    OPTIONAL{
        ?race :hasA ?qual.
        ?qual a :Qualifying.
        ?qualiPart a :QualifPartecipation;
                   :partecipatedInQualif ?qual;
                   :hasDriver ?driver;
                   :hasPosition ?qualiPos.
    }.
}
GROUP BY ?driversName
ORDER BY DESC (?numOfPoles)
LIMIT 1
```

## Name of the F1 driver with the highest number of pole position achieved

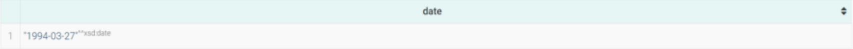
```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?driversName (COUNT(?driversName) as ?numOfPoles) WHERE {
    ?qualiPart a :QualifPartecipation;
               :hasPosition "1"^^xsd:integer;
               :hasDriver ?driver.
    ?driver a :Driver;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?driversName).
}
GROUP BY (?driversName)
ORDER BY DESC (?numOfPoles)
LIMIT 1
```



| | driversName | numOfPoles |
|---|---|---|
| 1 | "Lewis Hamilton" | "107"^^xsd:integer |

## Date of the first qualifying in the dataset

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?date WHERE {
    ?qualiPart a :QualifPartecipation;
               :partecipatedInQualif ?quali.
    ?quali a :Qualifying.
    ?race a :Race;
          :hasA ?quali;
          :hasDate ?date.
}
ORDER BY (?date)
LIMIT 1
```



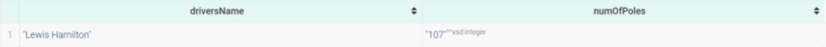
| | date |
|---|---|
| 1 | "1994-03-27"^^xsd:date |

## Name of the F1 driver with the highest number of pole position achieved, counted as number of times his starting grid position was 1. Also, printed the qualifying result in those races, to see if qualifying result is pole position or not (for penalties or other stuff)

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?driversName (COUNT(?driversName) as ?numOfPoles)
(GROUP_CONCAT((?qualiPos) ; separator=",") AS ?qualifPositions) WHERE {
    ?racePart a :RacePartecipation;
              :partecipatedInRace ?race;
              :hasStartingGridPosition "1"^^xsd:integer;
              :hasDriver ?driver.
    ?race a :Race.
    ?driver a :Driver;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?driversName).
    OPTIONAL{
        ?race :hasA ?qual.
        ?qual a :Qualifying.
        ?qualiPart a :QualifPartecipation;
               :partecipatedInQualif ?qual;
               :hasDriver ?driver;
               :hasPosition ?qualiPos.
    }.
}
GROUP BY ?driversName
ORDER BY DESC (?numOfPoles)
LIMIT 1
```

## Name of the F1 driver with the highest number of pole position achieved

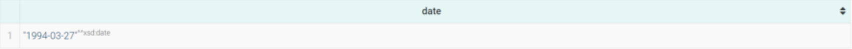
```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?driversName (COUNT(?driversName) as ?numOfPoles) WHERE {
    ?qualiPart a :QualifPartecipation;
               :hasPosition "1"^^xsd:integer;
               :hasDriver ?driver.
    ?driver a :Driver;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?driversName).
}
GROUP BY (?driversName)
ORDER BY DESC (?numOfPoles)
LIMIT 1
```



| | driversName | numOfPoles |
|---|---|---|
| 1 | "Lewis Hamilton" | "107"^^xsd:integer |

## Date of the first qualifying in the dataset

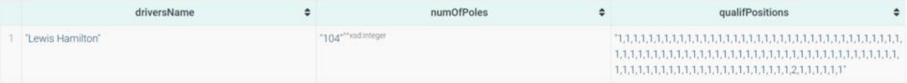
```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?date WHERE {
    ?qualiPart a :QualifPartecipation;
               :partecipatedInQualif ?quali.
    ?quali a :Qualifying.
    ?race a :Race;
          :hasA ?quali;
          :hasDate ?date.
}
ORDER BY (?date)
LIMIT 1
```



| | date |
|---|---|
| 1 | "1994-03-27"^^xsd:date |

## Name of the F1 driver with the highest number of pole position achieved, counted as number of times his starting grid position was 1. Also, printed the qualifying result in those races, to see if qualifying result is pole position or not (for penalties or other stuff)

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?driversName (COUNT(?driversName) as ?numOfPoles)
(GROUP_CONCAT((?qualiPos) ; separator=",") AS ?qualifPositions) WHERE {
    ?racePart a :RacePartecipation;
              :partecipatedInRace ?race;
              :hasStartingGridPosition "1"^^xsd:integer;
              :hasDriver ?driver.
    ?race a :Race.
    ?driver a :Driver;
            :hasForename ?fname;
            :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?driversName).
    OPTIONAL{
        ?race :hasA ?qual.
        ?qual a :Qualifying.
        ?qualiPart a :QualifPartecipation;
               :partecipatedInQualif ?qual;
               :hasDriver ?driver;
               :hasPosition ?qualiPos.
    }.
}
GROUP BY ?driversName
ORDER BY DESC (?numOfPoles)
LIMIT 1
```



| | driversName | numOfPoles | qualifPositions |
|---|---|---|---|
| 1 | "Lewis Hamilton" | "104"^^xsd:integer | "1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1" |

# Are drivers who won the championship after 2000 older than those who won it before 2000, on average?

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
ASK WHERE{
    {
        SELECT (AVG(?age) as ?avgOldAge) WHERE{
            ?stand :hasRace ?race ;
                   :hasDriver ?driv ;
                   :hasTotalPosition ?pos .
            ?driv :hasDateOfBirth ?dob .
            ?race :hasDate ?raceDate ;
                  :inSeason ?outerSeason ;
                  :hasRound ?outerRound .
            ?outerSeason :hasYear ?outerYear .
            BIND((YEAR(?raceDate) - YEAR(?dob)) AS ?age)
            FILTER(?outerRound = ?maxRound && ?outerYear = ?year && ?pos = 1)
            {
                SELECT ?year (MAX(?round) AS ?maxRound) WHERE {
                    ?race :inSeason ?season ;
                          :hasRound ?round .
                    ?season :hasYear ?year .
                    FILTER(?year < 2000)
                }
                GROUP BY ?year
            }
        }
    }
    {
        SELECT (AVG(?age) as ?avgRecentAge) WHERE{
            ?stand :hasRace ?race ;
                   :hasDriver ?driv ;
                   :hasTotalPosition ?pos .
            ?driv :hasDateOfBirth ?dob .
            ?race :hasDate ?raceDate ;
                  :inSeason ?outerSeason ;
                  :hasRound ?outerRound .
            ?outerSeason :hasYear ?outerYear .
            BIND((YEAR(?raceDate) - YEAR(?dob)) AS ?age)
            FILTER(?outerRound = ?maxRound && ?outerYear = ?year && ?pos = 1)
            {
                SELECT ?year (MAX(?round) AS ?maxRound) WHERE {
                    ?race :inSeason ?season ;
                          :hasRound ?round .
                    ?season :hasYear ?year .
                    FILTER(?year >= 2000)
                }
                GROUP BY ?year
            }
        }
    }
    FILTER(?avgRecentAge > ?avgOldAge)
}
```

# Are drivers who won the championship after 2000 older than those who won it before 2000, on average?

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
ASK WHERE{
    {
        SELECT (AVG(?age) as ?avgOldAge) WHERE{
            ?stand :hasRace ?race ;
                   :hasDriver ?driv ;
                   :hasTotalPosition ?pos .
            ?driv :hasDateOfBirth ?dob .
            ?race :hasDate ?raceDate ;
                  :inSeason ?outerSeason ;
                  :hasRound ?outerRound .
            ?outerSeason :hasYear ?outerYear .
            BIND((YEAR(?raceDate) - YEAR(?dob)) AS ?age)
            FILTER(?outerRound = ?maxRound && ?outerYear = ?year && ?pos = 1)
            {
                SELECT ?year (MAX(?round) AS ?maxRound) WHERE {
                    ?race :inSeason ?season ;
                          :hasRound ?round .
                    ?season :hasYear ?year .
                    FILTER(?year < 2000)
                }
                GROUP BY ?year
            }
        }
    }
    {
        SELECT (AVG(?age) as ?avgRecentAge) WHERE{
            ?stand :hasRace ?race ;
                   :hasDriver ?driv ;
                   :hasTotalPosition ?pos .
            ?driv :hasDateOfBirth ?dob .
            ?race :hasDate ?raceDate ;
                  :inSeason ?outerSeason ;
                  :hasRound ?outerRound .
            ?outerSeason :hasYear ?outerYear .
            BIND((YEAR(?raceDate) - YEAR(?dob)) AS ?age)
            FILTER(?outerRound = ?maxRound && ?outerYear = ?year && ?pos = 1)
            {
                SELECT ?year (MAX(?round) AS ?maxRound) WHERE {
                    ?race :inSeason ?season ;
                          :hasRound ?round .
                    ?season :hasYear ?year .
                    FILTER(?year >= 2000)
                }
                GROUP BY ?year
            }
        }
    }
    FILTER(?avgRecentAge > ?avgOldAge)
}
```

NO

## Every championship winner driver, ordered by season

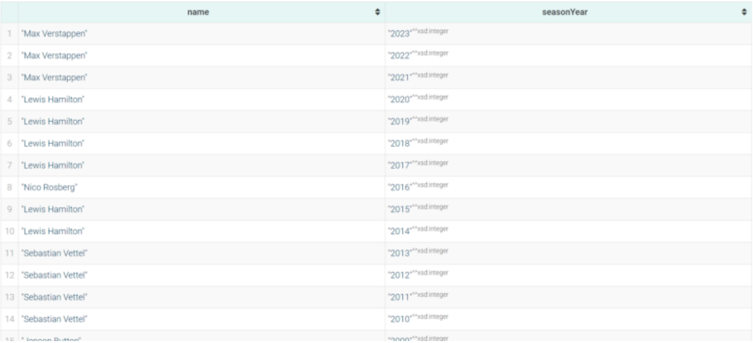
```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?name (?outerYear AS ?seasonYear) WHERE{
    ?stand :hasRace ?race ;
           :hasDriver ?driv ;
           :hasTotalPosition ?pos .
    ?driv a :Driver;
          :hasForename ?fname;
          :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
    ?race :hasDate ?raceDate ;
          :inSeason ?outerSeason ;
          :hasRound ?outerRound .
    ?outerSeason :hasYear ?outerYear .
    FILTER(?outerRound = ?maxRound && ?outerYear = ?year && ?pos = 1)
    {
        SELECT ?year (MAX(?round) AS ?maxRound) WHERE {
            ?race :inSeason ?season ;
                  :hasRound ?round .
            ?season :hasYear ?year .
        }
        GROUP BY ?year
    }
}
ORDER BY DESC (?seasonYear)
```

## Championship winning drivers who won more than 2 championships, ordered by number of championships won

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?name (COUNT(?name) AS ?numOfWins) WHERE{
    ?stand :hasRace ?race ;
           :hasDriver ?driv ;
           :hasTotalPosition ?pos .
    ?driv a :Driver;
          :hasForename ?fname;
          :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
    ?race :hasDate ?raceDate ;
          :inSeason ?outerSeason ;
          :hasRound ?outerRound .
    ?outerSeason :hasYear ?outerYear .
    FILTER(?outerRound = ?maxRound && ?outerYear = ?year && ?pos = 1)
    {
        SELECT ?year (MAX(?round) AS ?maxRound) WHERE {
            ?race :inSeason ?season ;
                  :hasRound ?round .
            ?season :hasYear ?year .
        }
        GROUP BY ?year
    }
}
GROUP BY ?name
HAVING (?numOfWins > 2)
ORDER BY DESC (?numOfWins)
```

## Every championship winner driver, ordered by season

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?name (?outerYear AS ?seasonYear) WHERE{
    ?stand :hasRace ?race ;
           :hasDriver ?driv ;
           :hasTotalPosition ?pos .
    ?driv a :Driver;
          :hasForename ?fname;
          :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
    ?race :hasDate ?raceDate ;
          :inSeason ?outerSeason ;
          :hasRound ?outerRound .
    ?outerSeason :hasYear ?outerYear .
    FILTER(?outerRound = ?maxRound && ?outerYear = ?year && ?pos = 1)
    {
        SELECT ?year (MAX(?round) AS ?maxRound) WHERE {
            ?race :inSeason ?season ;
                  :hasRound ?round .
            ?season :hasYear ?year .
        }
        GROUP BY ?year
    }
}
ORDER BY DESC (?seasonYear)
```



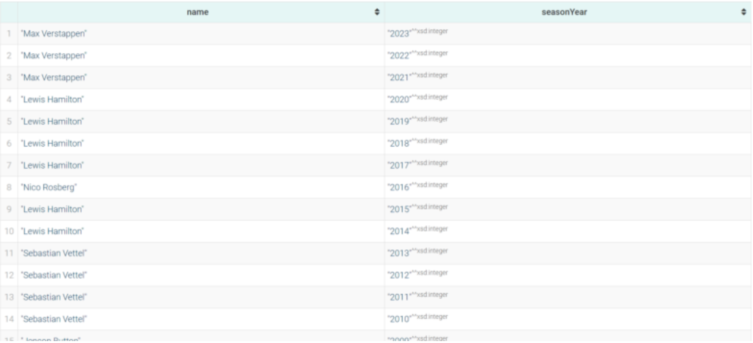
| | name | seasonYear |
|---|---|---|
| 1 | "Max Verstappen" | "2023"^^xsd:integer |
| 2 | "Max Verstappen" | "2022"^^xsd:integer |
| 3 | "Max Verstappen" | "2021"^^xsd:integer |
| 4 | "Lewis Hamilton" | "2020"^^xsd:integer |
| 5 | "Lewis Hamilton" | "2019"^^xsd:integer |
| 6 | "Lewis Hamilton" | "2018"^^xsd:integer |
| 7 | "Lewis Hamilton" | "2017"^^xsd:integer |
| 8 | "Nico Rosberg" | "2016"^^xsd:integer |
| 9 | "Lewis Hamilton" | "2015"^^xsd:integer |
| 10 | "Lewis Hamilton" | "2014"^^xsd:integer |
| 11 | "Sebastian Vettel" | "2013"^^xsd:integer |
| 12 | "Sebastian Vettel" | "2012"^^xsd:integer |
| 13 | "Sebastian Vettel" | "2011"^^xsd:integer |
| 14 | "Sebastian Vettel" | "2010"^^xsd:integer |
| 15 | "Jenson Button" | "2009"^^xsd:integer |

## Championship winning drivers who won more than 2 championships, ordered by number of championships won

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?name (COUNT(?name) AS ?numOfWins) WHERE{
    ?stand :hasRace ?race ;
           :hasDriver ?driv ;
           :hasTotalPosition ?pos .
    ?driv a :Driver;
          :hasForename ?fname;
          :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
    ?race :hasDate ?raceDate ;
          :inSeason ?outerSeason ;
          :hasRound ?outerRound .
    ?outerSeason :hasYear ?outerYear .
    FILTER(?outerRound = ?maxRound && ?outerYear = ?year && ?pos = 1)
    {
        SELECT ?year (MAX(?round) AS ?maxRound) WHERE {
            ?race :inSeason ?season ;
                  :hasRound ?round .
            ?season :hasYear ?year .
        }
        GROUP BY ?year
    }
}
GROUP BY ?name
HAVING (?numOfWins > 2)
ORDER BY DESC (?numOfWins)
```

## Every championship winner driver, ordered by season

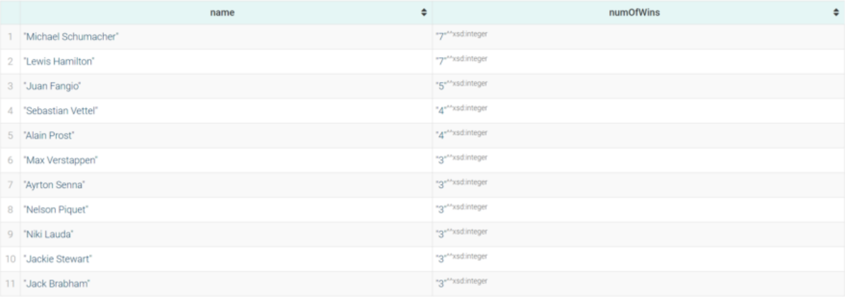
```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?name (?outerYear AS ?seasonYear) WHERE{
    ?stand :hasRace ?race ;
           :hasDriver ?driv ;
           :hasTotalPosition ?pos .
    ?driv a :Driver;
          :hasForename ?fname;
          :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
    ?race :hasDate ?raceDate ;
          :inSeason ?outerSeason ;
          :hasRound ?outerRound .
    ?outerSeason :hasYear ?outerYear .
    FILTER(?outerRound = ?maxRound && ?outerYear = ?year && ?pos = 1)
    {
        SELECT ?year (MAX(?round) AS ?maxRound) WHERE {
            ?race :inSeason ?season ;
                  :hasRound ?round .
            ?season :hasYear ?year .
        }
        GROUP BY ?year
    }
}
ORDER BY DESC (?seasonYear)
```



| | name | seasonYear |
|---|---|---|
| 1 | "Max Verstappen" | "2023"^^xsd:integer |
| 2 | "Max Verstappen" | "2022"^^xsd:integer |
| 3 | "Max Verstappen" | "2021"^^xsd:integer |
| 4 | "Lewis Hamilton" | "2020"^^xsd:integer |
| 5 | "Lewis Hamilton" | "2019"^^xsd:integer |
| 6 | "Lewis Hamilton" | "2018"^^xsd:integer |
| 7 | "Lewis Hamilton" | "2017"^^xsd:integer |
| 8 | "Nico Rosberg" | "2016"^^xsd:integer |
| 9 | "Lewis Hamilton" | "2015"^^xsd:integer |
| 10 | "Lewis Hamilton" | "2014"^^xsd:integer |
| 11 | "Sebastian Vettel" | "2013"^^xsd:integer |
| 12 | "Sebastian Vettel" | "2012"^^xsd:integer |
| 13 | "Sebastian Vettel" | "2011"^^xsd:integer |
| 14 | "Sebastian Vettel" | "2010"^^xsd:integer |
| 15 | "Jenson Button" | "2009"^^xsd:integer |

## Championship winning drivers who won more than 2 championships, ordered by number of championships won

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?name (COUNT(?name) AS ?numOfWins) WHERE{
    ?stand :hasRace ?race ;
           :hasDriver ?driv ;
           :hasTotalPosition ?pos .
    ?driv a :Driver;
          :hasForename ?fname;
          :hasSurname ?sname.
    BIND(CONCAT(?fname, " ", ?sname) AS ?name).
    ?race :hasDate ?raceDate ;
          :inSeason ?outerSeason ;
          :hasRound ?outerRound .
    ?outerSeason :hasYear ?outerYear .
    FILTER(?outerRound = ?maxRound && ?outerYear = ?year && ?pos = 1)
    {
        SELECT ?year (MAX(?round) AS ?maxRound) WHERE {
            ?race :inSeason ?season ;
                  :hasRound ?round .
            ?season :hasYear ?year .
        }
        GROUP BY ?year
    }
}
GROUP BY ?name
HAVING (?numOfWins > 2)
ORDER BY DESC (?numOfWins)
```



| | name | numOfWins |
|---|---|---|
| 1 | "Michael Schumacher" | "7"^^xsd:integer |
| 2 | "Lewis Hamilton" | "7"^^xsd:integer |
| 3 | "Juan Fangio" | "5"^^xsd:integer |
| 4 | "Sebastian Vettel" | "4"^^xsd:integer |
| 5 | "Alain Prost" | "4"^^xsd:integer |
| 6 | "Max Verstappen" | "3"^^xsd:integer |
| 7 | "Ayrton Senna" | "3"^^xsd:integer |
| 8 | "Nelson Piquet" | "3"^^xsd:integer |
| 9 | "Niki Lauda" | "3"^^xsd:integer |
| 10 | "Jackie Stewart" | "3"^^xsd:integer |
| 11 | "Jack Brabham" | "3"^^xsd:integer |

## Winners of driver's championship and constructor's championship for each season.

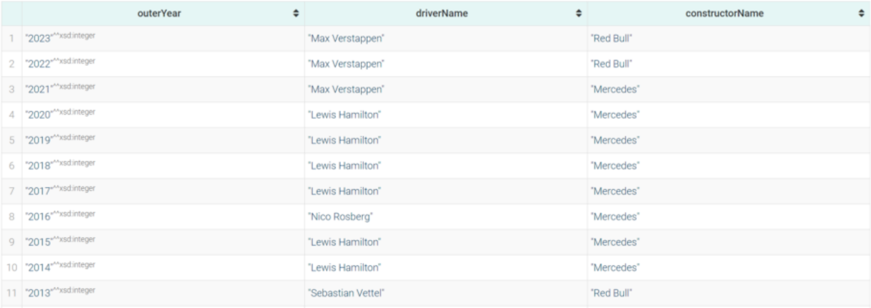
```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?outerYear ?driverName ?constructorName WHERE{
    ?driverStand :hasTotalPosition ?pos ;
                 :hasRace ?race ;
                 :hasDriver ?driver .
    ?constStand :hasTotalPosition ?pos ;
                :hasRace ?race ;
                :hasConstructor ?const .
    ?race :hasRound ?outerRound ;
          :inSeason ?season .
    ?season :hasYear ?outerYear .
    ?driver :hasForename ?fname ;
            :hasSurname ?sname .
    ?const :hasName ?constructorName .
    BIND(CONCAT(?fname, " ", ?sname) AS ?driverName).
    FILTER(?outerRound = ?maxRound && ?outerYear = ?year && ?pos = 1)
    {
        SELECT ?year (MAX(?round) AS ?maxRound) WHERE {
            ?race :inSeason ?season ;
                  :hasRound ?round .
            ?season :hasYear ?year .
        }
        GROUP BY ?year
    }
}
ORDER BY DESC (?outerYear)
```

## Drivers who gained the most positions during a race after 2000

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?driverName ?raceName ?year ?positionsGained WHERE{
    ?part :hasStartingGridPosition ?grid ;
          :partecipatedInRace ?race ;
          :hasDriver ?driver ;
          :hasPosition ?pos .
    ?driver :hasForename ?fname ;
            :hasSurname ?sname .
    ?race :inSeason ?season ;
          :hasName ?raceName .
    ?season :hasYear ?year .
    FILTER(?year >= 2000)
    BIND((?grid - ?pos) AS ?positionsGained)
    BIND(CONCAT(?fname, " ", ?sname) AS ?driverName)
}
ORDER BY DESC (?positionsGained)
LIMIT 50
```

## Winners of driver's championship and constructor's championship for each season.

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?outerYear ?driverName ?constructorName WHERE{
    ?driverStand :hasTotalPosition ?pos ;
                 :hasRace ?race ;
                 :hasDriver ?driver .
    ?constStand :hasTotalPosition ?pos ;
                :hasRace ?race ;
                :hasConstructor ?const .
    ?race :hasRound ?outerRound ;
          :inSeason ?season .
    ?season :hasYear ?outerYear .
    ?driver :hasForename ?fname ;
            :hasSurname ?sname .
    ?const :hasName ?constructorName .
    BIND(CONCAT(?fname, " ", ?sname) AS ?driverName).
    FILTER(?outerRound = ?maxRound && ?outerYear = ?year && ?pos = 1)
    {
        SELECT ?year (MAX(?round) AS ?maxRound) WHERE {
            ?race :inSeason ?season ;
                  :hasRound ?round .
            ?season :hasYear ?year .
        }
        GROUP BY ?year
    }
}
ORDER BY DESC (?outerYear)
```



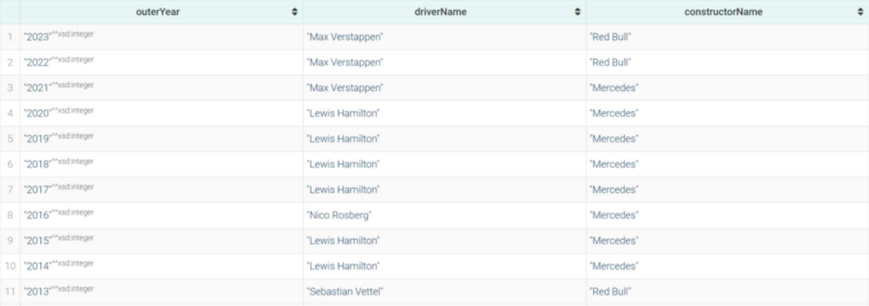
| | outerYear | driverName | constructorName |
|---|---|---|---|
| 1 | "2023"^^xsd:integer | "Max Verstappen" | "Red Bull" |
| 2 | "2022"^^xsd:integer | "Max Verstappen" | "Red Bull" |
| 3 | "2021"^^xsd:integer | "Max Verstappen" | "Mercedes" |
| 4 | "2020"^^xsd:integer | "Lewis Hamilton" | "Mercedes" |
| 5 | "2019"^^xsd:integer | "Lewis Hamilton" | "Mercedes" |
| 6 | "2018"^^xsd:integer | "Lewis Hamilton" | "Mercedes" |
| 7 | "2017"^^xsd:integer | "Lewis Hamilton" | "Mercedes" |
| 8 | "2016"^^xsd:integer | "Nico Rosberg" | "Mercedes" |
| 9 | "2015"^^xsd:integer | "Lewis Hamilton" | "Mercedes" |
| 10 | "2014"^^xsd:integer | "Lewis Hamilton" | "Mercedes" |
| 11 | "2013"^^xsd:integer | "Sebastian Vettel" | "Red Bull" |

## Drivers who gained the most positions during a race after 2000

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?driverName ?raceName ?year ?positionsGained WHERE{
    ?part :hasStartingGridPosition ?grid ;
          :partecipatedInRace ?race ;
          :hasDriver ?driver ;
          :hasPosition ?pos .
    ?driver :hasForename ?fname ;
            :hasSurname ?sname .
    ?race :inSeason ?season ;
          :hasName ?raceName .
    ?season :hasYear ?year .
    FILTER(?year >= 2000)
    BIND((?grid - ?pos) AS ?positionsGained)
    BIND(CONCAT(?fname, " ", ?sname) AS ?driverName)
}
ORDER BY DESC (?positionsGained)
LIMIT 50
```

## Winners of driver's championship and constructor's championship for each season.

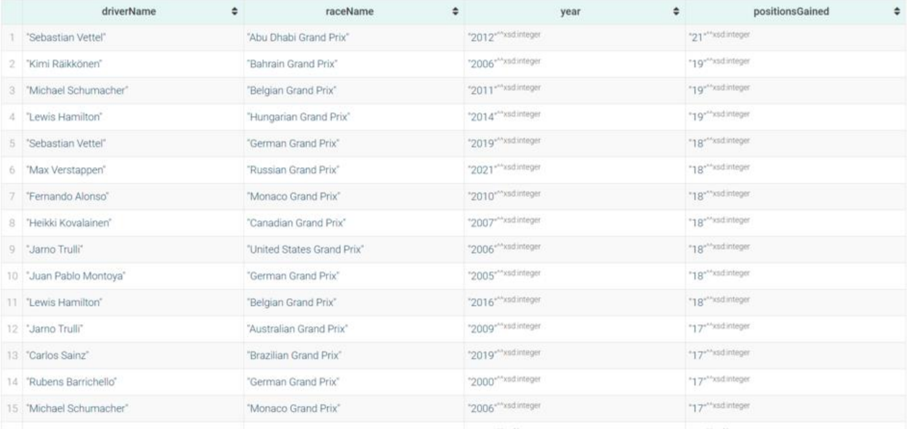
```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?outerYear ?driverName ?constructorName WHERE{
    ?driverStand :hasTotalPosition ?pos ;
                 :hasRace ?race ;
                 :hasDriver ?driver .
    ?constStand :hasTotalPosition ?pos ;
                :hasRace ?race ;
                :hasConstructor ?const .
    ?race :hasRound ?outerRound ;
          :inSeason ?season .
    ?season :hasYear ?outerYear .
    ?driver :hasForename ?fname ;
            :hasSurname ?sname .
    ?const :hasName ?constructorName .
    BIND(CONCAT(?fname, " ", ?sname) AS ?driverName).
    FILTER(?outerRound = ?maxRound && ?outerYear = ?year && ?pos = 1)
    {
        SELECT ?year (MAX(?round) AS ?maxRound) WHERE {
            ?race :inSeason ?season ;
                  :hasRound ?round .
            ?season :hasYear ?year .
        }
        GROUP BY ?year
    }
}
ORDER BY DESC (?outerYear)
```



| | outerYear | driverName | constructorName |
|---|---|---|---|
| 1 | "2023"^^xsd:integer | "Max Verstappen" | "Red Bull" |
| 2 | "2022"^^xsd:integer | "Max Verstappen" | "Red Bull" |
| 3 | "2021"^^xsd:integer | "Max Verstappen" | "Mercedes" |
| 4 | "2020"^^xsd:integer | "Lewis Hamilton" | "Mercedes" |
| 5 | "2019"^^xsd:integer | "Lewis Hamilton" | "Mercedes" |
| 6 | "2018"^^xsd:integer | "Lewis Hamilton" | "Mercedes" |
| 7 | "2017"^^xsd:integer | "Lewis Hamilton" | "Mercedes" |
| 8 | "2016"^^xsd:integer | "Nico Rosberg" | "Mercedes" |
| 9 | "2015"^^xsd:integer | "Lewis Hamilton" | "Mercedes" |
| 10 | "2014"^^xsd:integer | "Lewis Hamilton" | "Mercedes" |
| 11 | "2013"^^xsd:integer | "Sebastian Vettel" | "Red Bull" |

## Drivers who gained the most positions during a race after 2000

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?driverName ?raceName ?year ?positionsGained WHERE{
    ?part :hasStartingGridPosition ?grid ;
          :partecipatedInRace ?race ;
          :hasDriver ?driver ;
          :hasPosition ?pos .
    ?driver :hasForename ?fname ;
            :hasSurname ?sname .
    ?race :inSeason ?season ;
          :hasName ?raceName .
    ?season :hasYear ?year .
    FILTER(?year >= 2000)
    BIND((?grid - ?pos) AS ?positionsGained)
    BIND(CONCAT(?fname, " ", ?sname) AS ?driverName)
}
ORDER BY DESC (?positionsGained)
LIMIT 50
```



| | driverName | raceName | year | positionsGained |
|---|---|---|---|---|
| 1 | "Sebastian Vettel" | "Abu Dhabi Grand Prix" | "2012"^^xsd:integer | "21"^^xsd:integer |
| 2 | "Kimi Räikkönen" | "Bahrain Grand Prix" | "2006"^^xsd:integer | "19"^^xsd:integer |
| 3 | "Michael Schumacher" | "Belgian Grand Prix" | "2011"^^xsd:integer | "19"^^xsd:integer |
| 4 | "Lewis Hamilton" | "Hungarian Grand Prix" | "2014"^^xsd:integer | "19"^^xsd:integer |
| 5 | "Sebastian Vettel" | "German Grand Prix" | "2019"^^xsd:integer | "18"^^xsd:integer |
| 6 | "Max Verstappen" | "Russian Grand Prix" | "2021"^^xsd:integer | "18"^^xsd:integer |
| 7 | "Fernando Alonso" | "Monaco Grand Prix" | "2010"^^xsd:integer | "18"^^xsd:integer |
| 8 | "Heikki Kovalainen" | "Canadian Grand Prix" | "2007"^^xsd:integer | "18"^^xsd:integer |
| 9 | "Jarno Trulli" | "United States Grand Prix" | "2006"^^xsd:integer | "18"^^xsd:integer |
| 10 | "Juan Pablo Montoya" | "German Grand Prix" | "2005"^^xsd:integer | "18"^^xsd:integer |
| 11 | "Lewis Hamilton" | "Belgian Grand Prix" | "2016"^^xsd:integer | "18"^^xsd:integer |
| 12 | "Jarno Trulli" | "Australian Grand Prix" | "2009"^^xsd:integer | "17"^^xsd:integer |
| 13 | "Carlos Sainz" | "Brazilian Grand Prix" | "2019"^^xsd:integer | "17"^^xsd:integer |
| 14 | "Rubens Barrichello" | "German Grand Prix" | "2000"^^xsd:integer | "17"^^xsd:integer |
| 15 | "Michael Schumacher" | "Monaco Grand Prix" | "2006"^^xsd:integer | "17"^^xsd:integer |

## Drivers ordered by pole-to-win percentage (percentage of races won starting from the first position in grid)

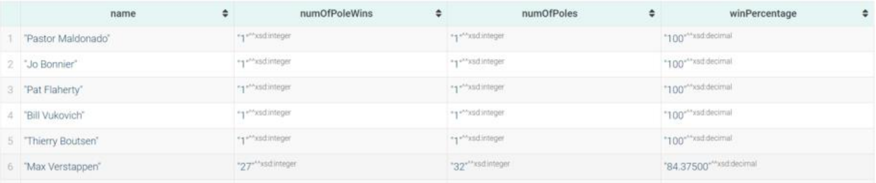
```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?name ?numOfPoleWins ?numOfPoles ?winPercentage WHERE{
    {
        SELECT ?name (COUNT(?name) AS ?numOfPoleWins) WHERE{
            ?racePart :partecipatedInRace ?race ;
                      :hasDriver ?driv ;
                      :hasStartingGridPosition ?startPos ;
                      :hasPosition ?pos .
            ?driv a :Driver;
                  :hasForename ?fname;
                  :hasSurname ?sname.
            BIND(CONCAT(?fname, " ", ?sname) AS ?name).
            FILTER(?startPos = 1 && ?pos = 1)
        }
        GROUP BY ?name
    }
    {
        SELECT ?name (COUNT(?name) AS ?numOfPoles) WHERE{
            ?racePart :partecipatedInRace ?race ;
                      :hasDriver ?driv ;
                      :hasStartingGridPosition ?startPos .
            ?driv a :Driver;
                  :hasForename ?fname;
                  :hasSurname ?sname.
            BIND(CONCAT(?fname, " ", ?sname) AS ?name).
            FILTER(?startPos = 1)
        }
        GROUP BY ?name
    }
    BIND((?numOfPoleWins / ?numOfPoles) * 100 AS ?winPercentage)
}
ORDER BY DESC (?winPercentage)
```

## Drivers who finished first in a race and were placed in the last quarter in the standing of the previous race.

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?driverName ?raceName ?round ?year WHERE{
    ?race :hasRound ?round ;
          :inSeason ?season ;
          :hasName ?raceName .
    ?part :partecipatedInRace ?race ;
          :hasPosition ?pos ;
          :hasDriver ?driver .
    ?driver :hasForename ?fname ;
            :hasSurname ?lname .
    ?season :hasYear ?year .
    BIND((?round-1) AS ?prevRound)
    BIND(CONCAT(?fname," ",?lname) AS ?driverName)
    FILTER(?pos = 1 && EXISTS{
            ?prevRace :hasRound ?prevRound ;
                      :inSeason ?season .
            ?stand :hasRace ?prevRace ;
                   :hasDriver ?driver ;
                   :hasTotalPosition ?totPos .
            FILTER (?totPos >= ((?maxPos/4)*3) && ?prevRace = ?raceMax)
            {
                SELECT ?raceMax (MAX(?totPos) AS ?maxPos) WHERE {
                    ?standing a :DriverStanding ;
                              :hasRace ?raceMax ;
                              :hasTotalPosition ?totPos .
                }
                GROUP BY ?raceMax
            }
        })
}
ORDER BY (?year)
```

## Drivers ordered by pole-to-win percentage (percentage of races won starting from the first position in grid)

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?name ?numOfPoleWins ?numOfPoles ?winPercentage WHERE{
    {
        SELECT ?name (COUNT(?name) AS ?numOfPoleWins) WHERE{
            ?racePart :partecipatedInRace ?race ;
                      :hasDriver ?driv ;
                      :hasStartingGridPosition ?startPos ;
                      :hasPosition ?pos .
            ?driv a :Driver;
                  :hasForename ?fname;
                  :hasSurname ?sname.
            BIND(CONCAT(?fname, " ", ?sname) AS ?name).
            FILTER(?startPos = 1 && ?pos = 1)
        }
        GROUP BY ?name
    }
    {
        SELECT ?name (COUNT(?name) AS ?numOfPoles) WHERE{
            ?racePart :partecipatedInRace ?race ;
                      :hasDriver ?driv ;
                      :hasStartingGridPosition ?startPos .
            ?driv a :Driver;
                  :hasForename ?fname;
                  :hasSurname ?sname.
            BIND(CONCAT(?fname, " ", ?sname) AS ?name).
            FILTER(?startPos = 1)
        }
        GROUP BY ?name
    }
    BIND((?numOfPoleWins / ?numOfPoles) * 100 AS ?winPercentage)
}
ORDER BY DESC (?winPercentage)
```



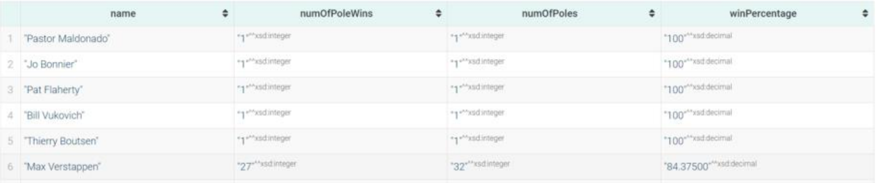
| | name | numOfPoleWins | numOfPoles | winPercentage |
|---|---|---|---|---|
| 1 | "Pastor Maldonado" | "1"^^xsd:integer | "1"^^xsd:integer | "100"^^xsd:decimal |
| 2 | "Jo Bonnier" | "1"^^xsd:integer | "1"^^xsd:integer | "100"^^xsd:decimal |
| 3 | "Pat Flaherty" | "1"^^xsd:integer | "1"^^xsd:integer | "100"^^xsd:decimal |
| 4 | "Bill Vukovich" | "1"^^xsd:integer | "1"^^xsd:integer | "100"^^xsd:decimal |
| 5 | "Thierry Boutsen" | "1"^^xsd:integer | "1"^^xsd:integer | "100"^^xsd:decimal |
| 6 | "Max Verstappen" | "27"^^xsd:integer | "32"^^xsd:integer | "84.37500"^^xsd:decimal |

## Drivers who finished first in a race and were placed in the last quarter in the standing of the previous race.

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?driverName ?raceName ?round ?year WHERE{
    ?race :hasRound ?round ;
          :inSeason ?season ;
          :hasName ?raceName .
    ?part :partecipatedInRace ?race ;
          :hasPosition ?pos ;
          :hasDriver ?driver .
    ?driver :hasForename ?fname ;
            :hasSurname ?lname .
    ?season :hasYear ?year .
    BIND((?round-1) AS ?prevRound)
    BIND(CONCAT(?fname," ",?lname) AS ?driverName)
    FILTER(?pos = 1 && EXISTS{
            ?prevRace :hasRound ?prevRound ;
                      :inSeason ?season .
            ?stand :hasRace ?prevRace ;
                   :hasDriver ?driver ;
                   :hasTotalPosition ?totPos .
            FILTER (?totPos >= ((?maxPos/4)*3) && ?prevRace = ?raceMax)
            {
                SELECT ?raceMax (MAX(?totPos) AS ?maxPos) WHERE {
                    ?standing a :DriverStanding ;
                              :hasRace ?raceMax ;
                              :hasTotalPosition ?totPos .
                }
                GROUP BY ?raceMax
            }
        })
}
ORDER BY (?year)
```

## Drivers ordered by pole-to-win percentage (percentage of races won starting from the first position in grid)

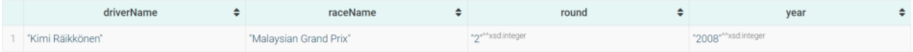
```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?name ?numOfPoleWins ?numOfPoles ?winPercentage WHERE{
    {
        SELECT ?name (COUNT(?name) AS ?numOfPoleWins) WHERE{
            ?racePart :partecipatedInRace ?race ;
                      :hasDriver ?driv ;
                      :hasStartingGridPosition ?startPos ;
                      :hasPosition ?pos .
            ?driv a :Driver;
                  :hasForename ?fname;
                  :hasSurname ?sname.
            BIND(CONCAT(?fname, " ", ?sname) AS ?name).
            FILTER(?startPos = 1 && ?pos = 1)
        }
        GROUP BY ?name
    }
    {
        SELECT ?name (COUNT(?name) AS ?numOfPoles) WHERE{
            ?racePart :partecipatedInRace ?race ;
                      :hasDriver ?driv ;
                      :hasStartingGridPosition ?startPos .
            ?driv a :Driver;
                  :hasForename ?fname;
                  :hasSurname ?sname.
            BIND(CONCAT(?fname, " ", ?sname) AS ?name).
            FILTER(?startPos = 1)
        }
        GROUP BY ?name
    }
    BIND((?numOfPoleWins / ?numOfPoles) * 100 AS ?winPercentage)
}
ORDER BY DESC (?winPercentage)
```



| | name | numOfPoleWins | numOfPoles | winPercentage |
|---|---|---|---|---|
| 1 | "Pastor Maldonado" | "1"^^xsd:integer | "1"^^xsd:integer | "100"^^xsd:decimal |
| 2 | "Jo Bonnier" | "1"^^xsd:integer | "1"^^xsd:integer | "100"^^xsd:decimal |
| 3 | "Pat Flaherty" | "1"^^xsd:integer | "1"^^xsd:integer | "100"^^xsd:decimal |
| 4 | "Bill Vukovich" | "1"^^xsd:integer | "1"^^xsd:integer | "100"^^xsd:decimal |
| 5 | "Thierry Boutsen" | "1"^^xsd:integer | "1"^^xsd:integer | "100"^^xsd:decimal |
| 6 | "Max Verstappen" | "27"^^xsd:integer | "32"^^xsd:integer | "84.37500"^^xsd:decimal |

## Drivers who finished first in a race and were placed in the last quarter in the standing of the previous race.

```
PREFIX : <https://www.dei.unipd.it/db2/groupProject/FASTianF1#>
SELECT ?driverName ?raceName ?round ?year WHERE{
    ?race :hasRound ?round ;
          :inSeason ?season ;
          :hasName ?raceName .
    ?part :partecipatedInRace ?race ;
          :hasPosition ?pos ;
          :hasDriver ?driver .
    ?driver :hasForename ?fname ;
            :hasSurname ?lname .
    ?season :hasYear ?year .
    BIND((?round-1) AS ?prevRound)
    BIND(CONCAT(?fname," ",?lname) AS ?driverName)
    FILTER(?pos = 1 && EXISTS{
            ?prevRace :hasRound ?prevRound ;
                      :inSeason ?season .
            ?stand :hasRace ?prevRace ;
                   :hasDriver ?driver ;
                   :hasTotalPosition ?totPos .
            FILTER (?totPos >= ((?maxPos/4)*3) && ?prevRace = ?raceMax)
            {
                SELECT ?raceMax (MAX(?totPos) AS ?maxPos) WHERE {
                    ?standing a :DriverStanding ;
                              :hasRace ?raceMax ;
                              :hasTotalPosition ?totPos .
                }
                GROUP BY ?raceMax
            }
        })
}
ORDER BY (?year)
```



| | driverName | raceName | round | year |
|---|---|---|---|---|
| 1 | "Kimi Räikkönen" | "Malaysian Grand Prix" | "2"^^xsd:integer | "2008"^^xsd:integer |

# Thank you!