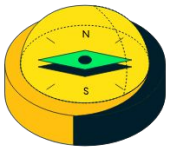
# F1 Base

Data Management project

Michele Nicoletti - 1886646
Lorenzo Pecorari - 1885161

# DM tool: MongoDB


- **NoSQL** DBMS
- collections and BSON documents
- flexible and fast
- advanced **query** operations
- Compass, mongosh, Atlas and pymongo

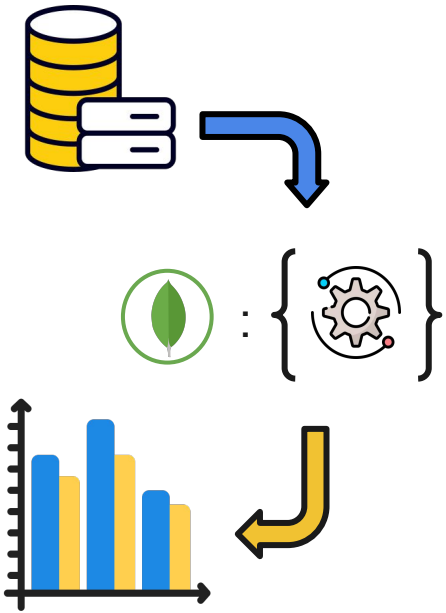
# Dataset and Data Modellation

- Kaggle: Formula 1 World Championship (1950 - 2024)
- circuits, drivers, constructors, races, standings, qualifyings, results and others


- csv files imported as collections with **referenced** model
- queries generating **embedded-modelled** data

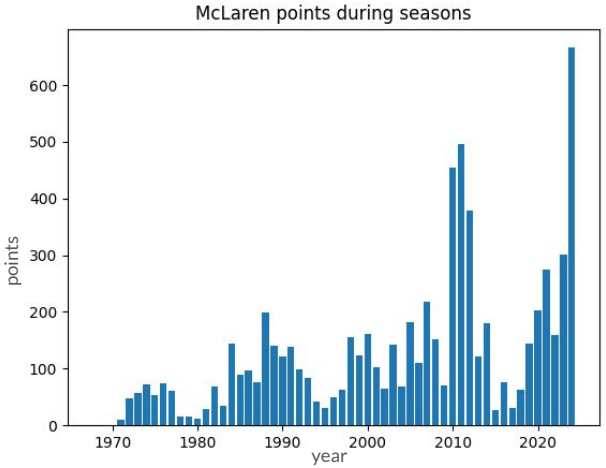
# Objectives of the project

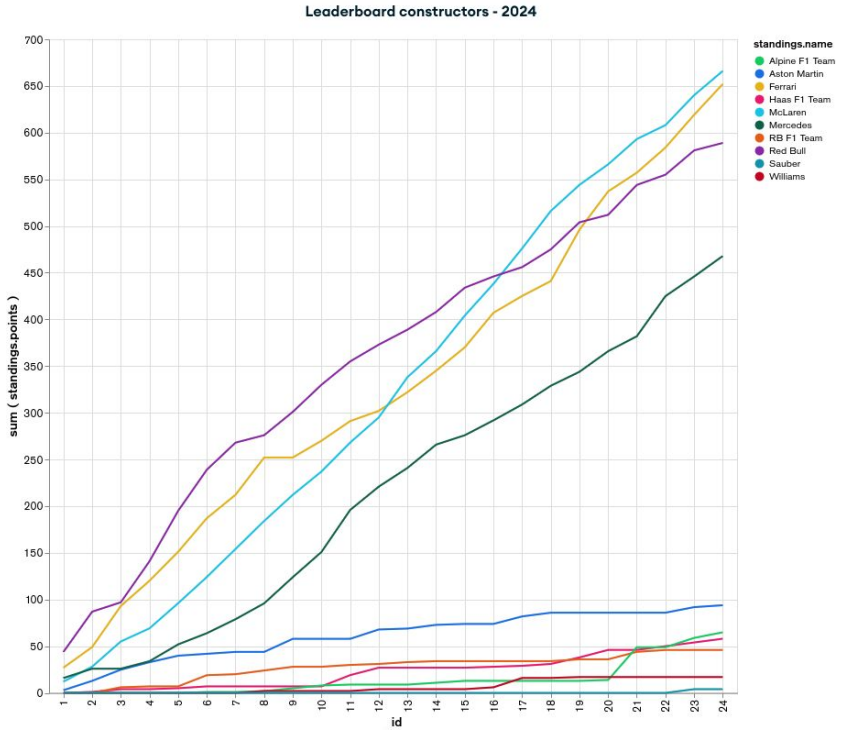
- dataset **exploration** to investigate:
  - constructors' performances
  - driver career performances
  - WDC and WCC wins among the seasons
  - failures and retirements
- Acquire knowledge about a **new** DM tool



# Constructors' performances

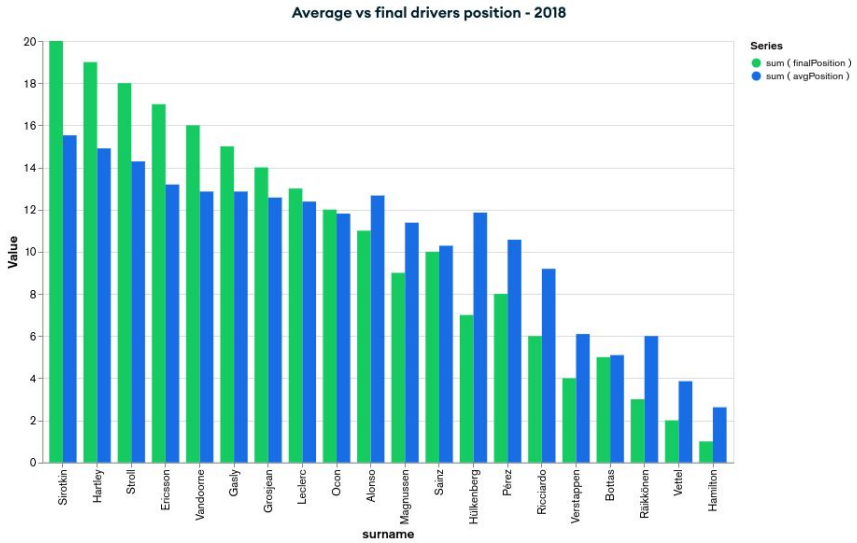
- leaderboard **evolution** during a season
- points-pit stops **comparison** of constructors during weekends
- team points **trend** of a team along seasons

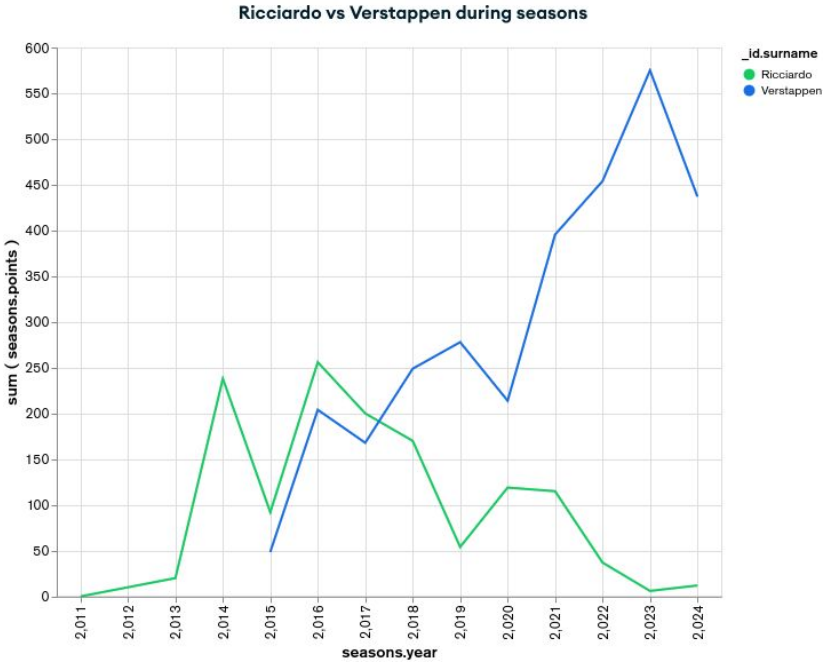


# Driver career performances

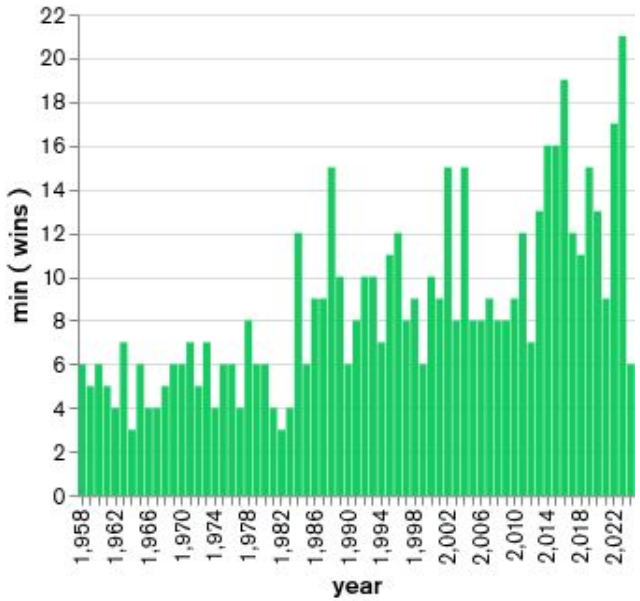
- average points scored in a **weekend**
- average points scored in a weekend per **constructors**
- **teammates** comparison in a season
- **final** position vs **average** position in a season



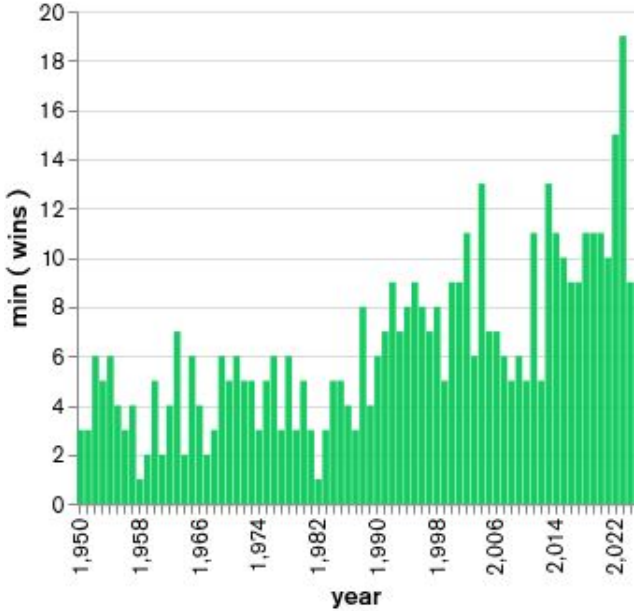



# WDC and WCC wins along the seasons

- number of wins of **constructors** champion along seasons

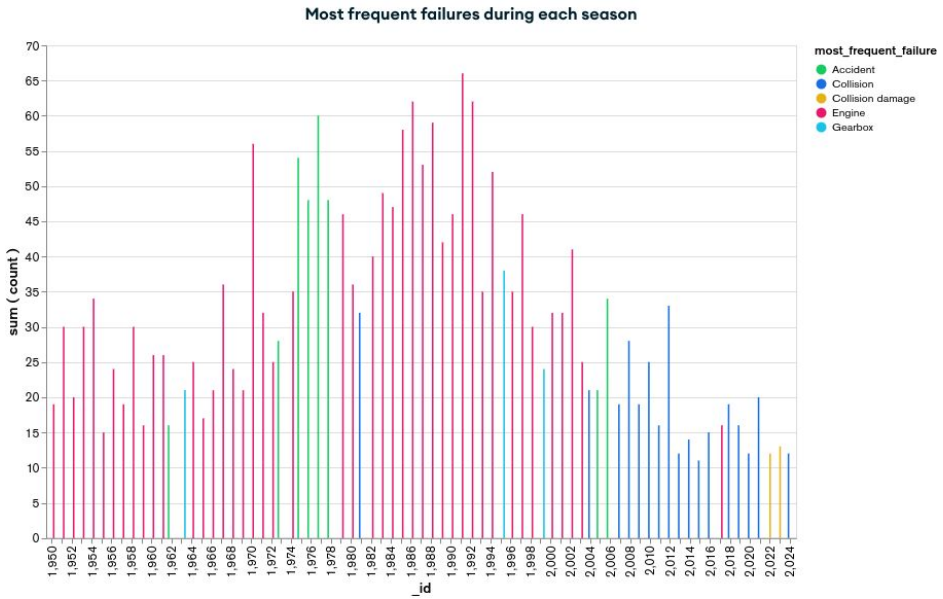


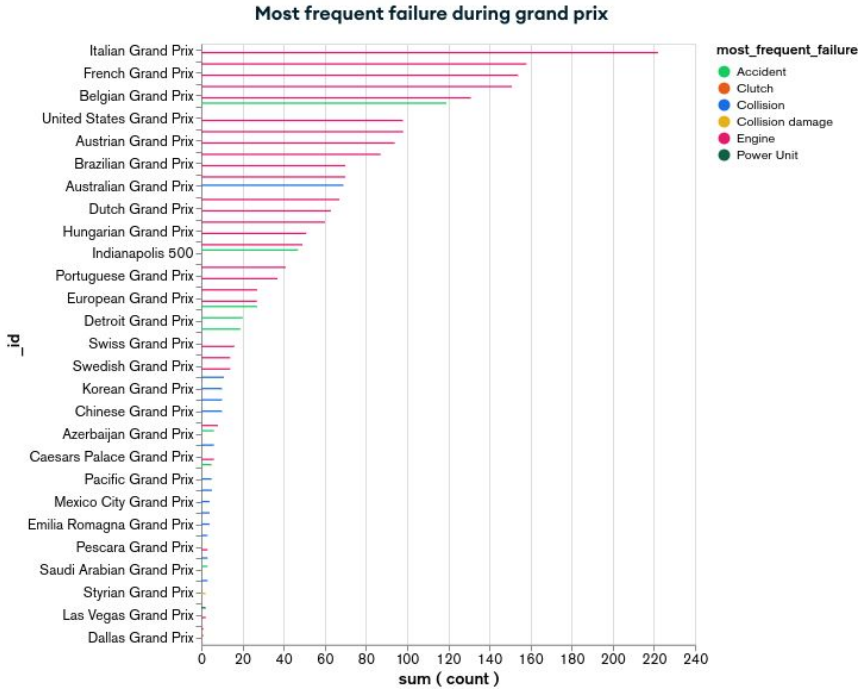

- number of wins of **drivers** champion during years



# Failures and retirements

- most frequent failures during all the **seasons**
- most frequent failure for each **grand prix**



# Query analysis: *“Driver champion wins”*

```
{
 $group: {
  _id: "$year",
  lastRound: {$max:"$round"}
 }
}
```

```
{$lookup:
 {from:"races",
  let:{year:"$_id",round:"$lastRound"},
  pipeline:[
    {$match:
     {$expr:{$and:
      [{$eq:["$year", "$$year"]},
       {$eq:["$round","$$round"]}]}}},
     {$project: {raceId: 1,year: 1}}
    ],
  as: "lastRace"}
}
```

```
{$unwind: "$lastRace"}
```

GROUP

LOOKUP

UNWIND

# Query analysis: *“Driver champion wins”*

```
{
 $project: {
  _id: 0,
  year: "$_id",
  driverId: "$standing.driverId",
  wins: "$standing.wins",
  name: "$driver.forename",
  surname: "$driver.surname"
 }
}
```

PROJECT

```
{
 $sort:{
   year:1
 }
}
```

SORT

# Indexes

- adopted for improving performances
- single and combined
- defined for **most used** collections in queries

ex: indexes for collection "*races*"

```
db.races.createIndex({year: 1})
db.races.createIndex({raceId: 1})
db.races.createIndex({year: 1, round: 1})
```

# Indexes: feedback on *“Driver champion wins”*

```
db.<COLLECTION>.aggregate(...).explain(“executionStats”)
```

```
{
  '$lookup': {
    from: 'driver_standings',
    as: 'standing',
    localField: 'lastRace.raceId',
    foreignField: 'raceId',
    let: {},
    pipeline: [
      { '$match': { position: { '$eq': 1 } } }
    ],
    unwinding: { preserveNullAndEmptyArrays: false }
  },
  totalDocsExamined: Long('2614725'),
  totalKeysExamined: Long('0'),
  collectionScans: Long('75'),
  indexesUsed: [],
  nReturned: Long('75'),
  executionTimeMillisEstimate: Long('715')
},
```

```
{
  '$lookup': {
    from: 'driver_standings',
    as: 'standing',
    localField: 'lastRace.raceId',
    foreignField: 'raceId',
    let: {},
    pipeline: [
      { '$match': { position: { '$eq': 1 } } }
    ],
    unwinding: { preserveNullAndEmptyArrays: false }
  },
  totalDocsExamined: Long('3192'),
  totalKeysExamined: Long('3192'),
  collectionScans: Long('0'),
  indexesUsed: [ 'raceId_1' ],
  nReturned: Long('75'),
  executionTimeMillisEstimate: Long('42')
},
```

# Data analysis through a script

- interaction between MongoDB and an application
- connection to the **cluster**
- querying a database of the cluster
- data analysis on JSON-like output

- command: `python3 main.py`