Data Analytics

Gender Diversity in Pop Music Charts

IronHack Paris
Matthieu Baillard

June, 2023

Table of content

Introduction

The music industry has long played a significant role in shaping cultural trends and reflecting societal values. Within this industry, pop music charts act as a barometer of popular taste, showcasing the most successful songs and artists of the time. Considering the well-documented influence and importance of representation, most notably on the youth, it is worth investigating how the place of women has evolved since the voices and contributions of little known classical music pioneers Hildegard von Bingen, Élisabeth Jacquet de la Guerre, Fanny Mendelssohn or Clara Schumann were largely ignored if not silenced from music history books.

The objective of our project will be to analyze gender diversity and unveil imbalances within the pop music charts. By examining the representation of male, female, and non-binary artists, we aim to spotlight the current state of the industry, and explore opportunities for fostering greater gender inclusivity.

Methodology

Our analysis is based on comprehensive data collected from an existing dataset on US charts, scraping from a French webzine, and fetching artist information through the API of two music websites. We will explore key metrics such as chart positions and longevity in the chart to gain insights into gender dynamics in the pop music industry.

Plan

- Project planning using ***Trello***
- Code management using ***Github***
- Scraping French weekly singles charts via **Python**
- Process artists from US & FR charts to identify main artists from featured artists
- Aggregate and deduplicate unique artists
- Fetch artist gender info from ***Musicbrainz API*** using Python
- Fetch artist bios from ***Last.fm API*** using Python
- Clean data & conduct EDA analysis with Python
- Design ERD and export data to **MySql**
- Conduct SQL queries
- Plan machine learning model to vectorize artist bios to recommend artists (closest female or non binary artist to the input artist).

Data and data sources

1. Billboard Hot 100 - USA, 1958-2022

***Billboard* Hot 100** is the music industry standard record chart in the United States for all-genre singles, published weekly by *Billboard* magazine since 1958. Chart rankings are based on retail and digital sales compiled by Nielsen Soundscan, radio airplay audience impressions as measured by Nielsen BDS, radio play, and online streaming activity provided by online music sources in the United States.

An existing dataset of complete historical charts data in .csv was downloaded from github.com/HipsterVizNinja/random-data/tree/main/Music/hot-100 which includes

336 295 rows x 13 columns

30 444 unique songs and 10 499 unique performers

chart_position	Position of the song for the given chart date
chart_date	Date that the chart was released
song	Name of the song
performer	Performer of the song - or *performers*, as we will see
song_id	Concatenation of song and performer to create a unique identifier
instance	Indicates how many times a song_id as returned to the chart after more than 1 week off the chart (See Mariah Carey - All I Want for Christmas is You for an example)

time_on_chart	Running count of weeks (all-time) a song_id has been on the chart
consecutive_weeks	For the given instance, how many weeks has the song_id been on the chart consecutively. A null value indicates the start of a new instance
previous_week	For the given instance, what was the chart_position for the previous week
peak_position	Indicates the all-time best/peak position for a song_id
worst_position	All-time lowest position for a song_id
chart_debut	Date of the first initial instance for a song_id
chart_url	URL to the chart on Billboard.com

2. France's Top 50 (1984-1998) & Top 100 (1998-2020)

We collected similar data for the French market, from official charts for sales of singles published on *chartsinfrance.net/charts/8444/singles.php*, which we web scraped using Python.

Source: *Pure Charts* is the official publisher of the official charts for singles sales in France, compiled by GFK Music for the SNEP (Syndicat National de l'Edition Phonographique)

152 551 rows x 7 columns

12 901 unique songs and 5 323 unique performers

Position	Position of the song for the given chart week *(1 to 50 until Feb. 20, 1998, then 1 to 100)*
Position Evolution	Evolution from next week's chart position (*e.g.* 'Entrée', '=', '+28', ...)
Artist	Performer(s) of the song (*e.g.* 'Peter et Sloan', 'Barbara Pravi', ...)
Song Title	Song title (*e.g.* 'Besoin de rien, envie de toi')
Year	Year of chart
Week	Week (*e.g.* Semaine du 26 octobre 1984)
URL Week ID	Week ID used as YYWW as used in weekly chart's URL (*e.g.* 8444, 2051)

3. MusicBrainz API

Our priority was to get the gender from the performers in the previously collected charts. We did so via ***Musicbrainz.org***, an open music encyclopedia that collects and shares music metadata. In addition to gender information, we fetched additional data through their ***API*** using Python, such as artist type (person, group or other), dates, area and tags (including musical genres).
Note: groups will not be considered in the current project, as they don't directly have a gender - though their individual members do, and so we might handle groups in a later iteration.

We managed to collect **13495 rows x 8 columns**,
down to 10754 unique artists after dropping duplicates

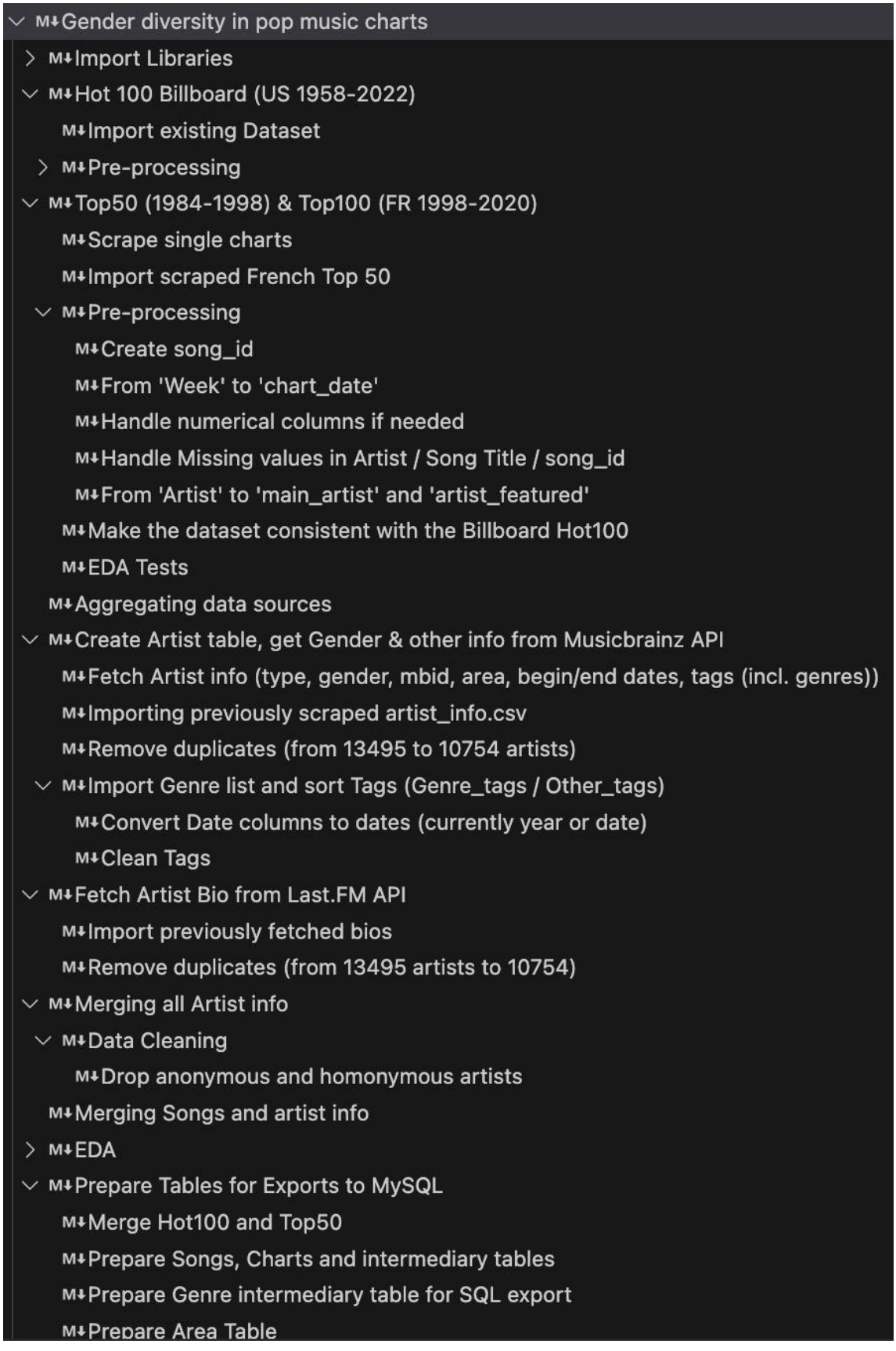
Artist	Artist name
MusicBrainz ID	Unique artist ID (e.g. 6655955b-1c1e-4bcb-84e4-81bcd9efab30)
Type	Type of Artist: Person, Group, Character, Orchestra, Other
Gender	Gender of Artist (Male / Female / Non-binary / Other / N.A.)
Area	Main geographic region of the Artist (country, state, city…)
Begin Date	Date of Birth or Creation of Group (Year or Date)
End Date	Date of Death or disbanding of a Group (Year or Date)
Tags	User-generated tags, including but not limited to musical genres

4. Last.fm API

Last but not least, we collected artist bios through Last.fm's API, via Python using previously collected unique 'MusicBrainz ID' to connect to the wanted artists.

Data collection and cleaning

[Python Notebook]



Exploratory data analysis

Global dataset Billboard Hot100

	chart_position	instance	time_on_chart	consecutive_weeks	previous_week	peak_position	worst_position
count	336295.000000	336295.000000	336295.000000	302987.000000	302987.000000	336295.000000	336295.000000
mean	50.499326	1.081384	9.223732	8.805952	47.561179	40.818579	80.675351
std	28.865716	0.394334	7.738961	7.386418	28.047132	29.342796	18.178729
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	25.500000	1.000000	4.000000	3.000000	23.000000	13.000000	74.000000
50%	50.000000	1.000000	7.000000	7.000000	47.000000	38.000000	86.000000
75%	75.000000	1.000000	13.000000	12.000000	71.000000	65.000000	94.000000
max	100.000000	15.000000	91.000000	89.000000	100.000000	100.000000	100.000000

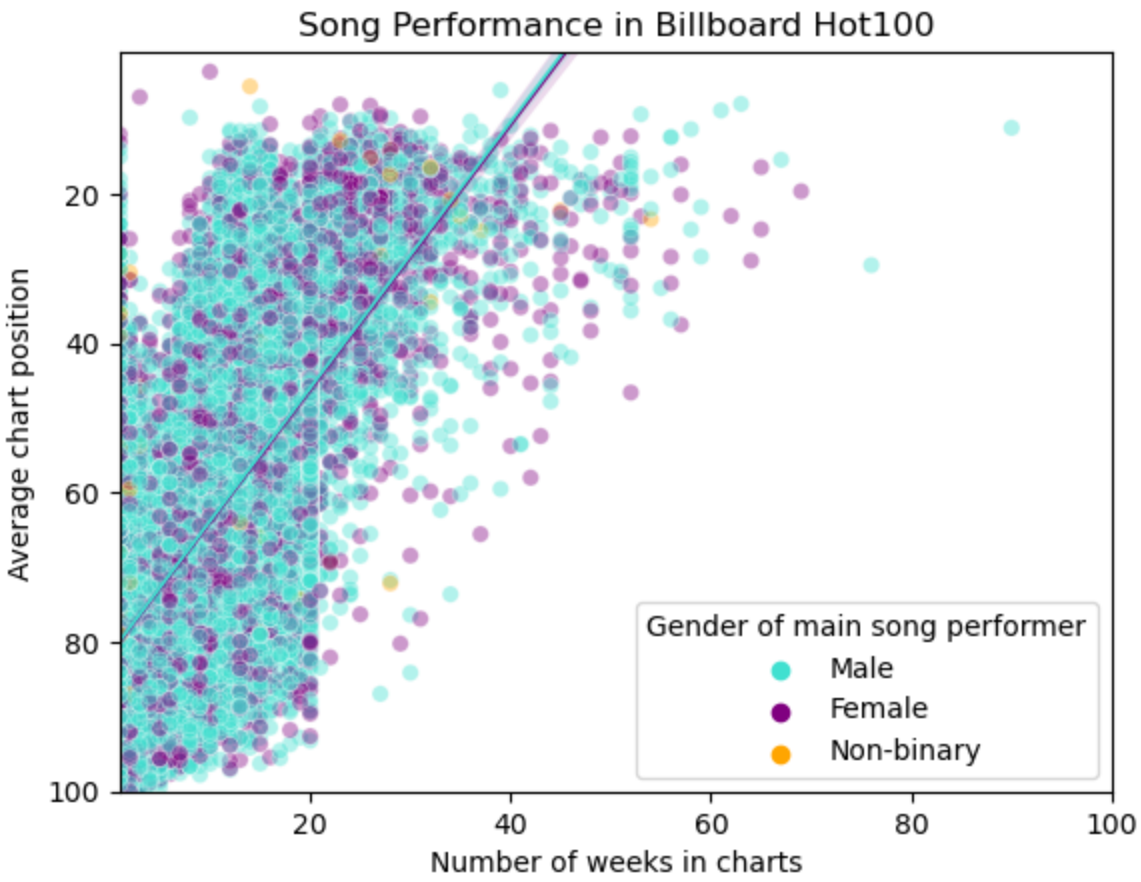
Billboard Hot100 filtered on songs by Female performers show average position in charts slightly higher *(note the reverse scale for chart positions, i.e. 1 is higher than 100)* than songs by their Male counterparts. The chart data by Female artists also shows a slightly higher variance.
But most notably, they are significantly less numerous in the charts.

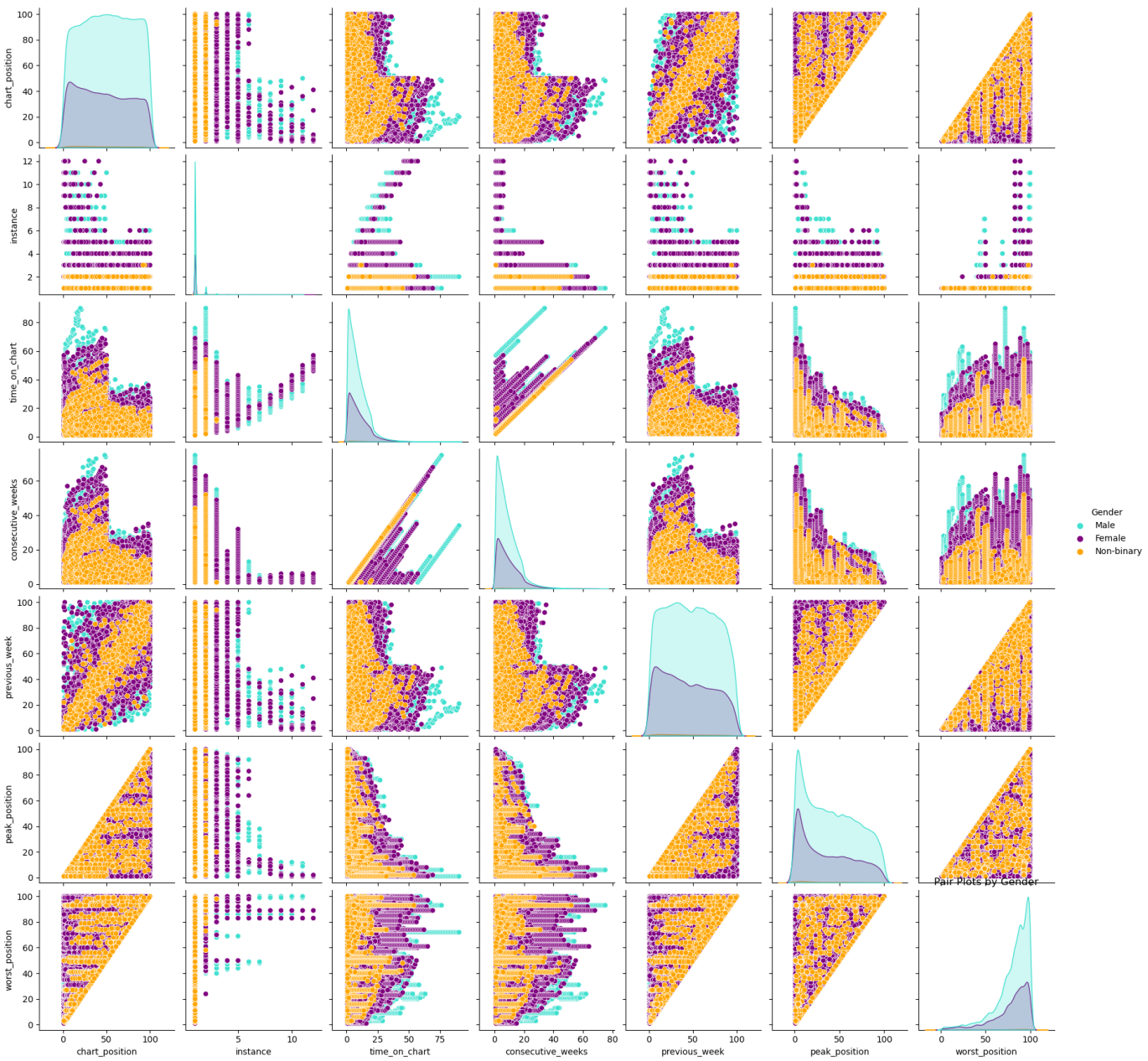
```
hot100_with_artist_info[hot100_with_artist_info['Gender'] == 'Female'].describe()
```
✓ 0.1s

	chart_position	instance	time_on_chart	consecutive_weeks	previous_week	peak_position	worst_position
count	55781.000000	55781.000000	55781.000000	50626.000000	50626.000000	55781.000000	55781.000000
mean	47.456571	1.107743	10.158316	9.691581	44.425651	36.753070	77.941091
std	29.344435	0.501709	8.423614	8.074999	28.316688	29.534451	20.656485
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	22.000000	1.000000	4.000000	4.000000	19.000000	9.000000	70.000000
50%	46.000000	1.000000	8.000000	8.000000	42.000000	31.000000	84.000000
75%	73.000000	1.000000	14.000000	14.000000	68.000000	61.000000	93.000000
max	100.000000	12.000000	69.000000	68.000000	100.000000	100.000000	100.000000

```
hot100_with_artist_info[hot100_with_artist_info['Gender'] == 'Male'].describe()
```
✓ 0.1s

	chart_position	instance	time_on_chart	consecutive_weeks	previous_week	peak_position	worst_position
count	130345.000000	130345.000000	130345.000000	116652.000000	116652.000000	130345.000000	130345.000000
mean	51.077464	1.090606	9.067383	8.687352	48.032927	41.296605	80.456535
std	28.528177	0.399970	7.554151	7.207376	27.726436	29.070435	18.752753
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	27.000000	1.000000	3.000000	3.000000	24.000000	15.000000	73.000000
50%	51.000000	1.000000	7.000000	7.000000	47.000000	39.000000	86.000000
75%	76.000000	1.000000	13.000000	12.000000	71.000000	65.000000	94.000000
max	100.000000	11.000000	90.000000	75.000000	100.000000	100.000000	100.000000


Song Performance in Billboard Hot100
Average chart position
Number of weeks in charts
20
40
60
80
100
Gender of main song performer
Male
Female
Non-binary


Pair Plots by Gender
chart_position
instance
time_on_chart
consecutive_weeks
previous_week
peak_position
worst_position
Gender
Male
Female
Non-binary

Database Type

We have opted for **MySQL** to store our project's data.

MySQL is an **open source** database management software that delivers a fast multi-user, and robust SQL (Structured Query Language) database server.

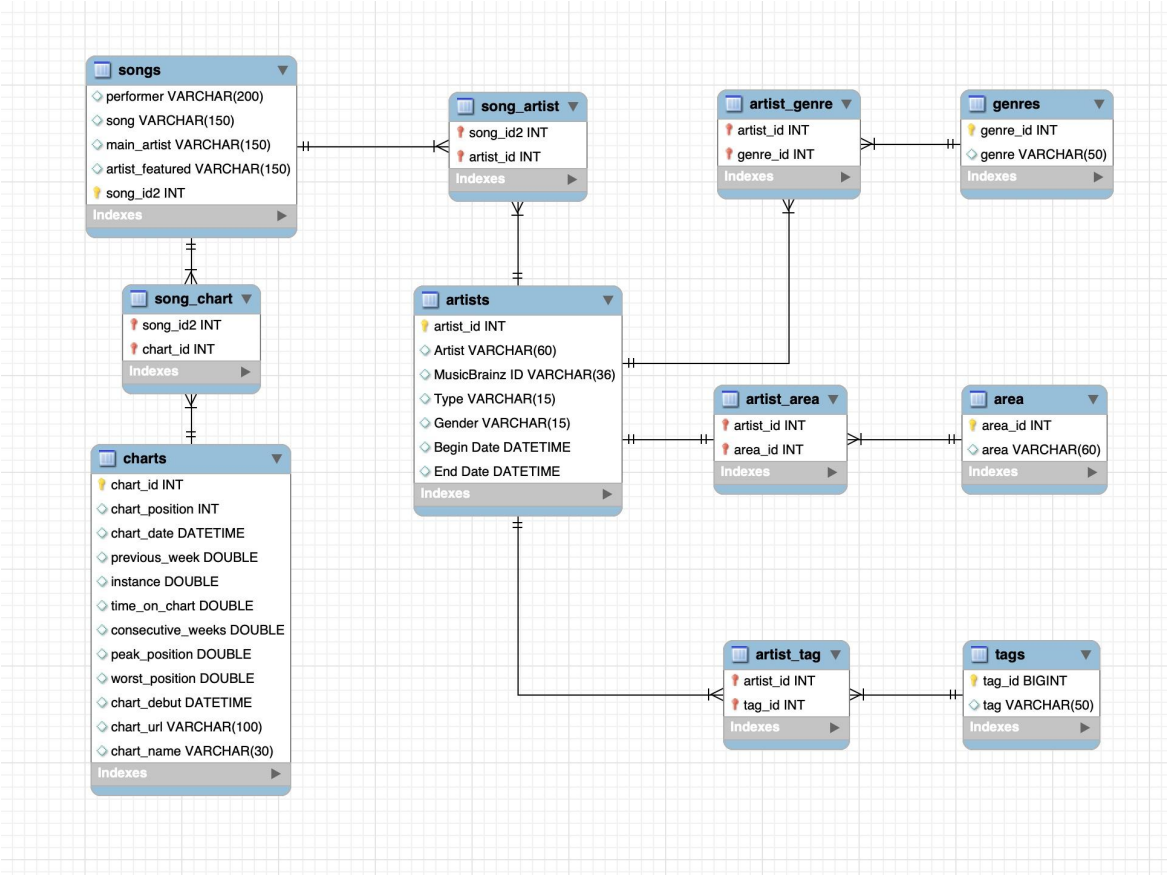
It is low-cost -*mostly free*-, highly flexible, and benefits from a large support community.

It is designed to manage structured, **relational data**, and therefore appropriate for our dataset.

It also allows for easy BI tools such as Tableau integration for analysis and visualization.

For less structured data, such as social media contents, NoSQL solutions such as MongoDB could have been a better-suited alternative.

ERD


songs
performer VARCHAR(200)
song VARCHAR(150)
main_artist VARCHAR(150)
artist_featured VARCHAR(150)
song_id2 INT
Indexes
song_artist
song_id2 INT
artist_id INT
Indexes
artist_genre
artist_id INT
genre_id INT
Indexes
genres
genre_id INT
genre VARCHAR(50)
Indexes
song_chart
song_id2 INT
chart_id INT
Indexes
artists
artist_id INT
Artist VARCHAR(60)
MusicBrainz ID VARCHAR(36)
Type VARCHAR(15)
Gender VARCHAR(15)
Begin Date DATETIME
End Date DATETIME
Indexes
artist_area
artist_id INT
area_id INT
Indexes
area
area_id INT
area VARCHAR(60)
Indexes
charts
chart_id INT
chart_position INT
chart_date DATETIME
previous_week DOUBLE
instance DOUBLE
time_on_chart DOUBLE
consecutive_weeks DOUBLE
peak_position DOUBLE
worst_position DOUBLE
chart_debut DATETIME
chart_url VARCHAR(100)
chart_name VARCHAR(30)
Indexes
artist_tag
artist_id INT
tag_id INT
Indexes
tags
tag_id BIGINT
tag VARCHAR(50)
Indexes

SQL Queries

```sql
-- AVERAGE CHART POSITION OF A SONG BY A FEMALE PERFORMER

SELECT AVG(charts.chart_position) AS mean_position
FROM charts
JOIN song_chart ON charts.chart_id = song_chart.chart_id
-- JOIN songs ON song_chart.song_id2 = songs.song_id2
JOIN song_artist ON song_chart.song_id2 = song_artist.song_id2
JOIN artists ON song_artist.artist_id = artists.artist_id
WHERE artists.gender = 'Female';
```

mean_position
45.9895

```sql
-- How many songs chart per artist on average (per decade)?

SELECT charts.chart_country,
       CONCAT(YEAR(charts.chart_date) DIV 10 * 10, 's') AS decade, artists.gender,
       ROUND(COUNT(DISTINCT song_chart.song_id2) / COUNT(DISTINCT(artists.artist_id)), 0) AS avg_song_count
FROM charts
JOIN song_chart ON charts.chart_id = song_chart.chart_id
JOIN song_artist ON song_chart.song_id2 = song_artist.song_id2
JOIN artists ON song_artist.artist_id = artists.artist_id
WHERE (artists.gender = 'Female' OR artists.gender = 'Male' OR artists.gender = 'Non-binary')
GROUP BY charts.chart_country, decade, artists.gender
ORDER BY decade, charts.chart_country, artists.gender;
```

	chart_country	decade	gender	avg_song_count
▶	US	1950s	Female	2
	US	1950s	Male	3
	US	1960s	Female	5
	US	1960s	Male	5
	US	1970s	Female	4
	US	1970s	Male	4
	FR	1980s	Female	2
	FR	1980s	Male	2
	US	1980s	Female	4
	US	1980s	Male	3
	FR	1990s	Female	3
	FR	1990s	Male	2
	US	1990s	Female	3
	US	1990s	Male	3
	FR	2000s	Female	3
	FR	2000s	Male	2
	FR	2000s	Non-binary	1
	US	2000s	Female	4
	US	2000s	Male	4
	US	2000s	Non-binary	9
	FR	2010s	Female	4
	FR	2010s	Male	4
	FR	2010s	Non-binary	4
	US	2010s	Female	4
	US	2010s	Male	6
	US	2010s	Non-binary	12
	FR	2020s	Female	2
	FR	2020s	Male	2
	FR	2020s	Non-binary	2
	US	2020s	Female	5
	US	2020s	Male	6
	US	2020s	Non-binary	10

```sql
SELECT charts.chart_country,
       -- Group by decades:
       CONCAT(YEAR(charts.chart_date) DIV 10 * 10, 's') AS decade,
       -- How many distinct artists in the charts of a known given gender?
       artists.gender, COUNT(DISTINCT(artists.artist_id)) AS artist_count,
       -- What is the average chart position of a song by an artist of a known given gender?
       ROUND(AVG(charts.chart_position), 0) AS mean_position,
       -- How long does a song by an artist of a given gender stay in the charts on average?
       ROUND(COUNT(charts.chart_id) / COUNT(DISTINCT song_chart.song_id2), 0) AS avg_weeks_on_chart,
       -- How many songs by an artist of a known given gender on an average weekly chart?
       ROUND(COUNT(charts.chart_id) / COUNT(DISTINCT charts.chart_date), 0) AS avg_songs_per_week,
       -- How many unique songs by gender artist that decade in each chart t
       COUNT(DISTINCT song_chart.song_id2) AS total_songs
FROM charts
JOIN song_chart ON charts.chart_id = song_chart.chart_id
JOIN song_artist ON song_chart.song_id2 = song_artist.song_id2
JOIN artists ON song_artist.artist_id = artists.artist_id
WHERE (artists.gender = 'Female' OR artists.gender = 'Male' OR artists.gender = 'Non-binary')
-- AND charts.chart_date >= '1998-02-27' -- If we wanted comparable results for France (Top50 before that date)
GROUP BY charts.chart_country, decade, artists.gender
ORDER BY decade, charts.chart_country, artists.gender
;
```

chart_country	decade	gender	artist_count	mean_position	avg_weeks_on_chart	avg_songs_per_week	total_songs
US	1950s	Female	37	53	7	9	89
US	1950s	Male	176	50	8	53	483
US	1960s	Female	186	52	7	12	875
US	1960s	Male	549	52	7	41	2908
US	1970s	Female	179	50	10	12	635
US	1970s	Male	517	50	9	35	1919
FR	1980s	Female	90	24	14	11	215
FR	1980s	Male	134	26	14	14	282
US	1980s	Female	172	49	13	16	612
US	1980s	Male	402	49	13	32	1320
FR	1990s	Female	173	34	13	11	434
FR	1990s	Male	284	32	12	15	650
US	1990s	Female	226	46	16	19	644
US	1990s	Male	356	51	14	26	954
FR	2000s	Female	251	48	16	22	706
FR	2000s	Male	421	50	15	28	976
FR	2000s	Non-binary	1	54	23	1	1
US	2000s	Female	201	46	15	21	729
US	2000s	Male	372	52	14	38	1371
US	2000s	Non-binary	2	42	9	2	17
FR	2010s	Female	288	48	10	21	1061
FR	2010s	Male	651	50	9	42	2422
FR	2010s	Non-binary	6	45	13	1	26
US	2010s	Female	187	45	12	18	778
US	2010s	Male	439	52	11	52	2429
US	2010s	Non-binary	5	42	12	2	62
FR	2020s	Female	95	48	6	27	207
FR	2020s	Male	234	47	4	43	464
FR	2020s	Non-binary	4	67	4	2	9
US	2020s	Female	83	46	9	21	375
US	2020s	Male	234	50	7	57	1307
US	2020s	Non-binary	5	57	4	2	50

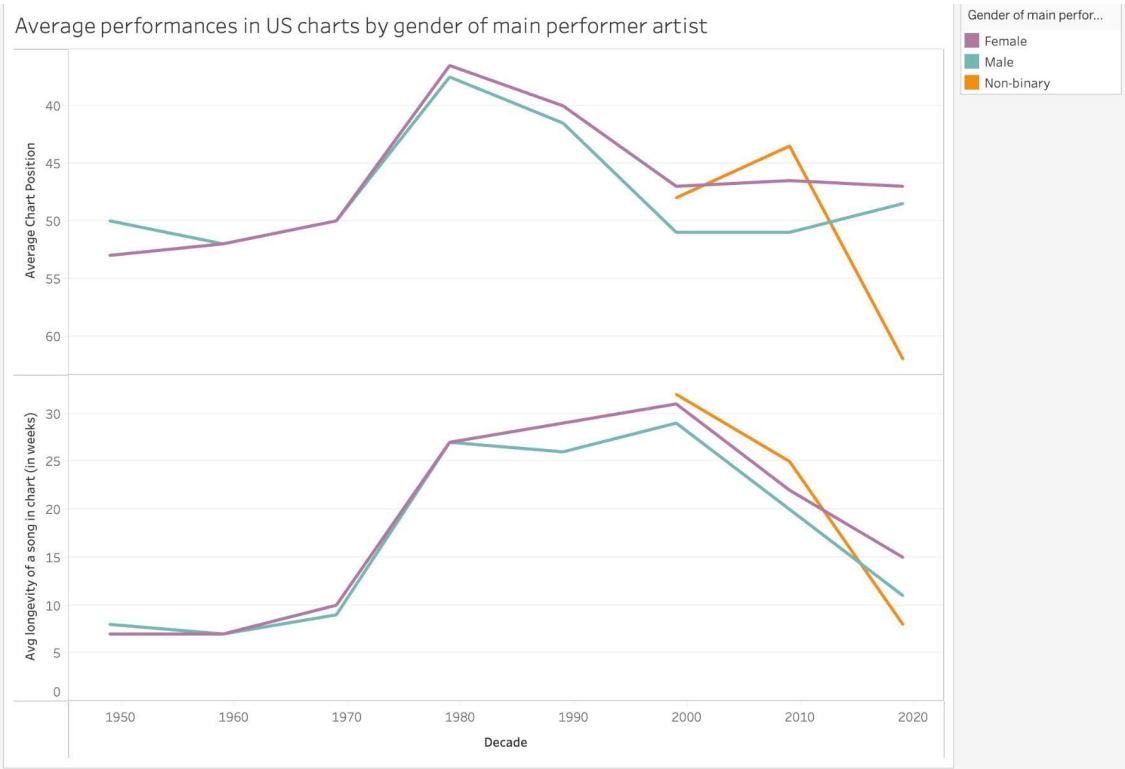
```sql
-- Who are the top 10 artists in number of songs that chart, with avg chart position?

SELECT artists.Artist,
       charts.chart_country,
       CONCAT(YEAR(charts.chart_date) DIV 10 * 10, 's') AS decade,
       artists.gender,
       COUNT(DISTINCT song_chart.song_id2) AS song_count,
       ROUND(AVG(charts.chart_position), 1) AS avg_chart_position,
       MIN(charts.chart_position) AS best_chart_position
FROM charts
JOIN song_chart ON charts.chart_id = song_chart.chart_id
JOIN song_artist ON song_chart.song_id2 = song_artist.song_id2
JOIN artists ON song_artist.artist_id = artists.artist_id
WHERE artists.gender IN ('Female', 'Male', 'Non-binary')
GROUP BY artists.Artist, charts.chart_country, decade, artists.gender
ORDER BY song_count DESC
LIMIT 10;
```

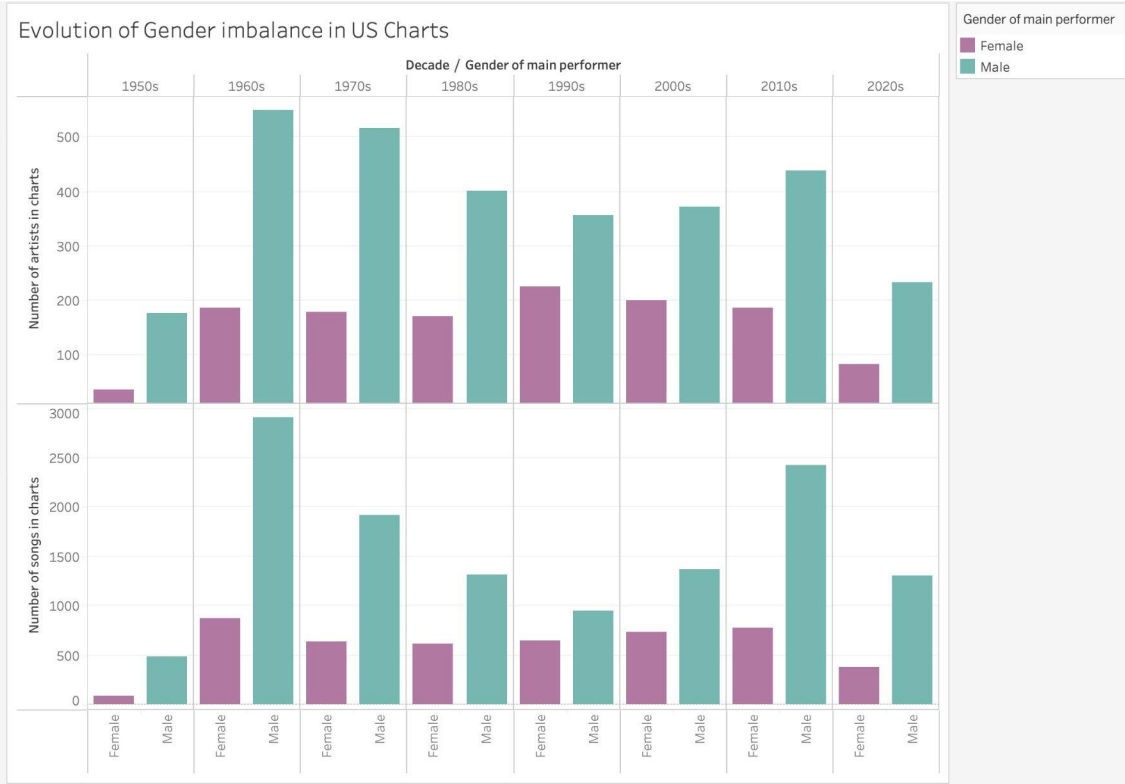
Artist	chart_country	decade	gender	song_count	avg_chart_positi...	best_chart_positi...
Drake	US	2010s	Male	136	44.9	1
Taylor Swift	US	2020s	Female	92	53.5	1
Taylor Swift	US	2010s	Female	72	38.5	1
Elvis Presley	US	1960s	Male	72	37.4	1
Jul	FR	2010s	Male	70	60.2	3
Drake	US	2020s	Male	66	39.6	1
YoungBoy Never Broke Again	US	2020s	Male	66	77.7	28
Lil Baby	US	2020s	Male	63	55.8	3
Johnny Hallyday	FR	2010s	Male	62	49.2	1
Ray Charles	US	1960s	Male	59	48.6	1

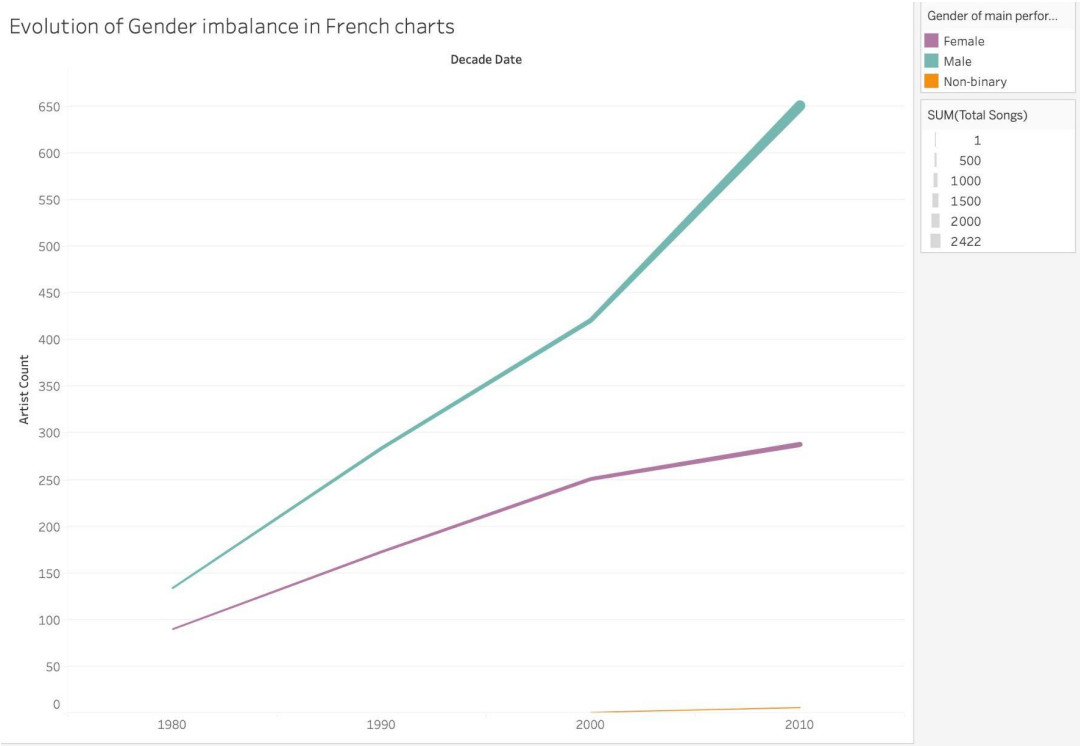
```sql
-- Who are the 10 artists who have remained at #1 for the largest number of weeks in the US?

SELECT artists.Artist, artists.gender,
COUNT(charts.chart_id) AS weeks_at_no1
FROM charts
JOIN song_chart ON charts.chart_id = song_chart.chart_id
JOIN song_artist ON song_chart.song_id2 = song_artist.song_id2
JOIN artists ON song_artist.artist_id = artists.artist_id
WHERE charts.chart_country = 'US' AND charts.chart_position = 1
    AND artists.gender IN ('Female', 'Male', 'Non-binary')
GROUP BY artists.Artist, artists.gender
ORDER BY artists.gender, weeks_at_no1 DESC
LIMIT 10;
```

Artist	gender	weeks_at_no1
Mariah Carey	Female	75
Usher	Male	47
Drake	Male	43
Rihanna	Female	43
Adele	Female	34
Katy Perry	Female	33
Madonna	Female	32
Michael Jackson	Male	31
Whitney Houston	Female	31
Taylor Swift	Female	27

Additional Visualization






Evolution of Gender imbalance in French charts
Decade Date
Artist Count
650
600
550
500
450
400
350
300
250
200
150
100
50
0
1980
1990
2000
2010
Gender of main perfor...
Female
Male
Non-binary
SUM(Total Songs)
1
500
1000
1500
2000
2422

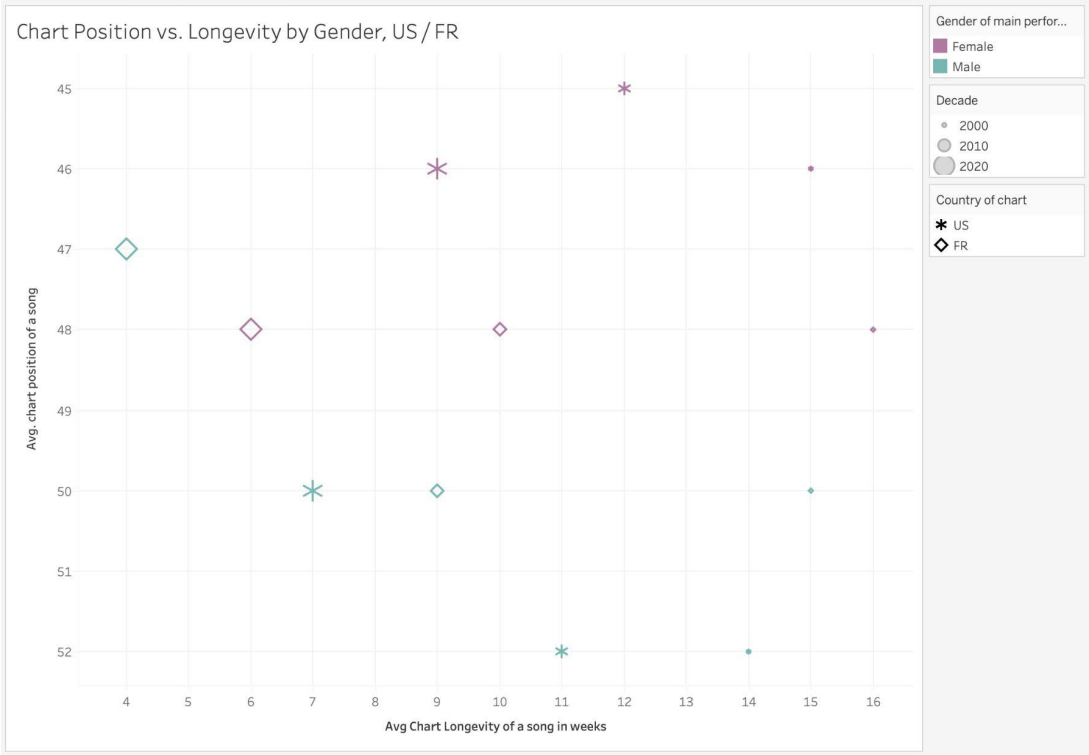


Chart Position vs. Longevity by Gender, US / FR
Avg. chart position of a song
45
46
47
48
49
50
51
52
4
5
6
7
8
9
10
11
12
13
14
15
16
Avg Chart Longevity of a song in weeks
Gender of main perfor...
Female
Male
Decade
2000
2010
2020
Country of chart
US
FR

Conclusion

The data confirms that there are huge disparities in the presence of women in pop music charts in the US and France. Even though the evolution has been positive for women through the end of the 20th century, the situation unfortunately tends to step back since the 2000's.
We could remain optimistic in observing that songs by female artists perform slightly better in the charts on average than those by male artists. They are driven though by a few megastars who should not overshadow the fact that women and gender minorities still have overall and by far access to a ridiculously small share of the mainstream pop music market.

Links

https://github.com/matthieubaillard/gender-diversity-in-music-charts
https://trello.com/b/PmFSf3XR/ironhack-final-project