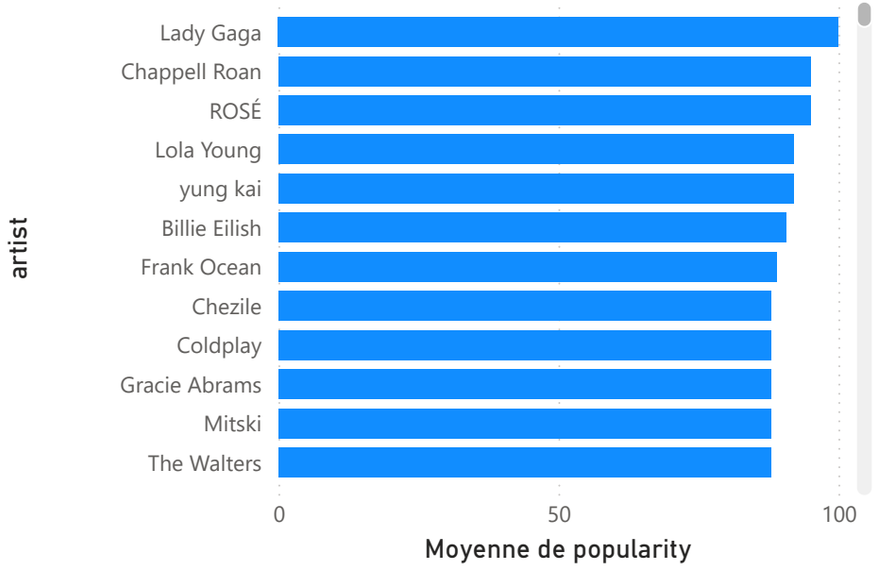
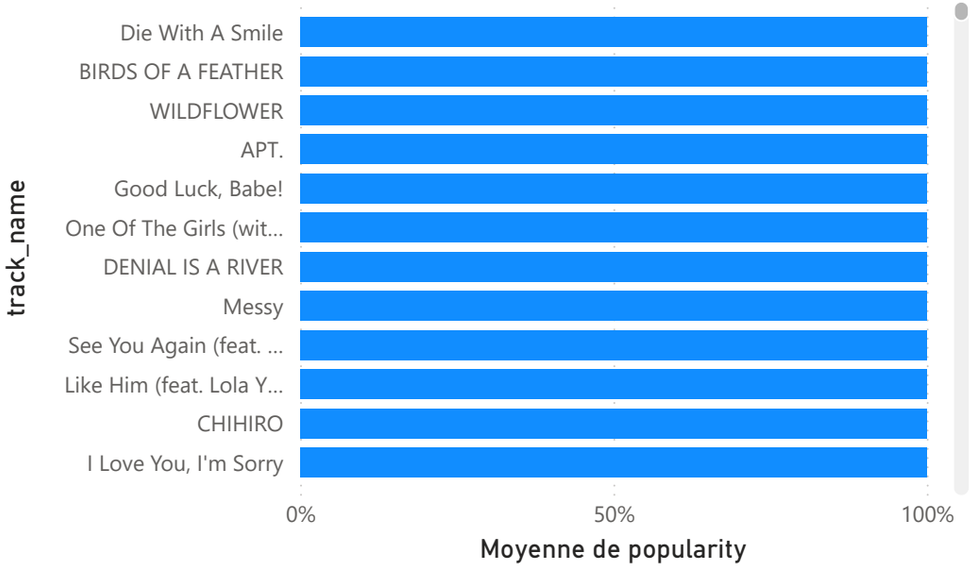
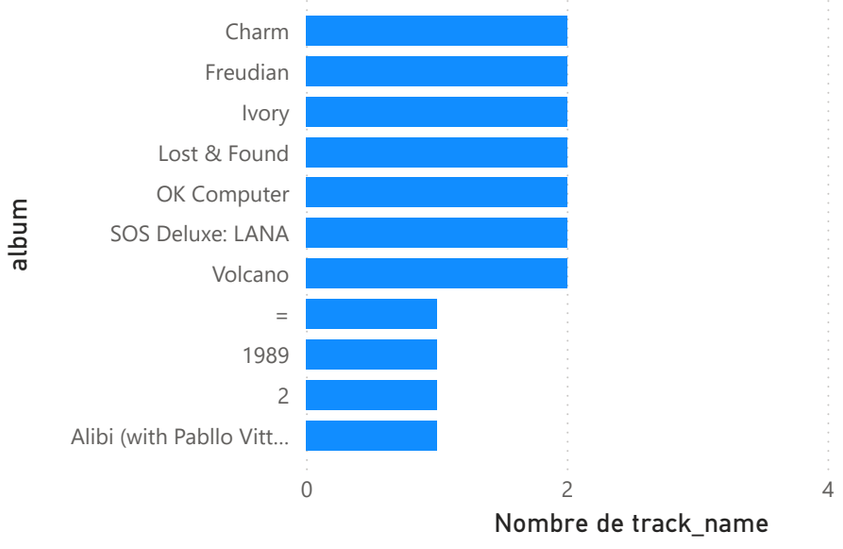
Moyenne de popularity par artist



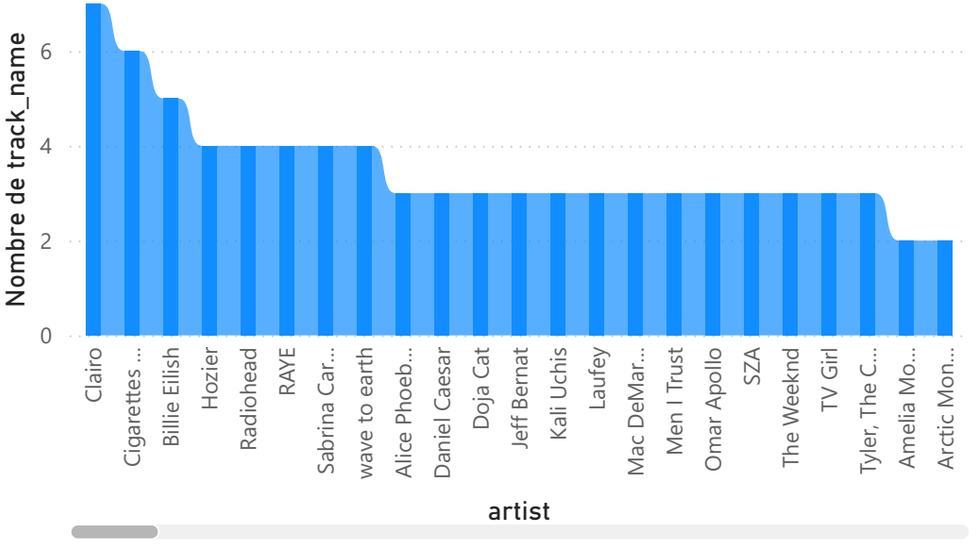
Moyenne de popularity par track_name



Nombre de track_name par album



Nombre de track_name par artist



Mesure rapide

```python
# 1. Import Pandas,Numpy,Seaborn,Matplotlib

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sqlite3

import time

# Étape 1 : Temps avant optimisation
start = time.time()

df = pd.read_csv('/workspaces/simple-/track_info.csv')
df
```

```
                            track_name              artist  \
0    See You Again (feat. Kali Uchis)  Tyler, The Creator
1                             Dark Red          Steve Lacy
2                         Pink + White         Frank Ocean
3                         Lovers Rock             TV Girl
4                  Something About You            Eyedress
..                                 ...                 ...
305                               2516             Luna Li
306                          Bamboléo         Gipsy Kings
307    Ojos Tristes (with The Marías)        Selena Gomez
308                          Tú Con Él        Frankie Ruiz
309                            Nothing         Bruno Major

                       album release_date  popularity  \
0                  Flower Boy   2017-07-21          92
1                    Dark Red   2017-02-20          86
2                      Blonde   2016-08-20          89
3                 French Exit   2014-06-05          88
4           Mulholland Drive   2021-08-27          84
..                        ...          ...         ...
305                   jams EP   2021-02-05          54
306              Gipsy Kings   1987-08-24          71
307  I Said I Love You First   2025-03-21          82
308      El Papá De La Salsa   2008-09-09          64
309  To Let A Good Thing Die   2020-06-05          75

                          artist_genre
0                                  NaN
1                                  NaN
2                                  NaN
3                                  NaN
4                                  NaN
..                                 ...
305                                NaN
```

```
306                              flamenco
307                                   pop
308  salsa, salsa romantica, merengue
309                                   NaN

[310 rows x 6 columns]
```

```
print("Shape of the dataset:",df.shape)
```

```
Shape of the dataset: (310, 6)
```

```
# 5. Missing values
```

```
df.isnull().sum()
```

```
track_name        0
artist            0
album             0
release_date      0
popularity        0
artist_genre    187
dtype: int64
```

```
# 4. Column details
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 310 entries, 0 to 309
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   track_name    310 non-null    object
 1   artist        310 non-null    object
 2   album         310 non-null    object
 3   release_date  310 non-null    object
 4   popularity    310 non-null    int64 
 5   artist_genre  123 non-null    object
dtypes: int64(1), object(5)
memory usage: 14.7+ KB
```

```
# 5. Statistics of the data
```

```
df.describe().transpose()
```

```
            count       mean        std  min   25%   50%    75%    max
popularity  310.0  66.648387  20.810804  0.0  58.0  72.0  81.75  100.0
```

```
import sqlite3
import pandas as pd
```

```python
# Connexion à la base de données SQLite
conn = sqlite3.connect("spotify.db")

import sqlite3
import pandas as pd

# 1. Charger le fichier CSV
df_csv = pd.read_csv('/workspaces/simple-/track_info.csv')

import time

start = time.time()

df_all = pd.read_sql_query("SELECT * FROM spotify", conn)

end = time.time()
temps_avant = end - start
print(f"Temps avant optimisation : {temps_avant:.4f} s")

# Code optimisé (lecture ciblée + agrégation)
start = time.time()
df_grouped = pd.read_sql_query("""
    SELECT artist, COUNT(track_name) as total_tracks
    FROM spotify
    GROUP BY artist
""", conn)
end = time.time()
temps_apres = end - start
print(f"Temps après optimisation : {temps_apres:.4f} s")

# Pourcentage d'amélioration
if temps_avant > 0:
    gain = ((temps_avant - temps_apres) / temps_avant) * 100
    print(f"Amélioration : {gain:.2f}%")
```

```
Temps avant optimisation : 0.0028 s
Temps après optimisation : 0.0011 s
Amélioration : 60.02%
```

```python
df_grouped = pd.read_sql_query("""
    SELECT artist, COUNT(track_name) as total_tracks
    FROM spotify
    GROUP BY artist
""", conn)
print(df_grouped)
```

```
            artist  total_tracks
0        A$AP Ferg             1
1       AP Dhillon             1
2             ATFC             1
3      Addison Rae             1
```

```
4              Air             1
..             ...           ...
228         juju<3             1
229         slchld             1
230    thenightsky             1
231  wave to earth             4
232       yung kai             1

[233 rows x 2 columns]
```

```
conn = sqlite3.connect("spotify.db")
cursor = conn.cursor()

with open("/workspaces/simple-/data.sql", "r") as f:
    sql_script = f.read()

cursor.executescript(sql_script)
conn.commit()

df_csv.to_sql("spotify", conn, if_exists="replace", index=False)
df = pd.read_sql_query("SELECT count(*) AS total_lignes FROM
spotify;", conn)
print(df)
```

```
   total_lignes
0           310
```

```
import sqlite3
import pandas as pd

conn = sqlite3.connect("spotify.db")

df1 = pd.read_sql_query("SELECT count(distinct album) AS nb_albums
FROM spotify;", conn)
print("nombre d'albums :", df1.iloc[0, 0])
```

```
nombre d'albums : 292
```

```
query1 = """
SELECT COUNT(track_name) AS total_no_counts, artist
FROM spotify
GROUP BY artist
ORDER BY total_no_counts DESC;
"""
df1 = pd.read_sql_query(query1, conn)
print(df1.head())
```

```
   total_no_counts                artist
0                7                Clairo
1                6  Cigarettes After Sex
2                5         Billie Eilish
```

```
3                4         wave to earth
4                4     Sabrina Carpenter
```

```
query2 = """
SELECT artist, AVG(popularity) AS average_popularity
FROM spotify
GROUP BY artist
ORDER BY average_popularity DESC;
"""
df2 = pd.read_sql_query(query2, conn)
print(df2.head())
```

```
          artist  average_popularity
0      Lady Gaga               100.0
1           ROSÉ                95.0
2  Chappell Roan                95.0
3       yung kai                92.0
4     Lola Young                92.0
```

```
query3 = """
SELECT track_name, artist, popularity
FROM spotify
ORDER BY popularity DESC
LIMIT 10;
"""
df3 = pd.read_sql_query(query3, conn)
print(df3)
```

```
                                        track_name               artist
\
0                                Die With A Smile           Lady Gaga

1                              BIRDS OF A FEATHER       Billie Eilish

2                                      WILDFLOWER       Billie Eilish

3                                Good Luck, Babe!       Chappell Roan

4                                            APT.                ROSÉ

5  One Of The Girls (with JENNIE, Lily Rose Depp)          The Weeknd

6                See You Again (feat. Kali Uchis)  Tyler, The Creator

7                                            blue            yung kai

8                                           Messy          Lola Young

9                            DENIAL IS A RIVER             Doechii
```

```
   popularity
0         100
1          99
2          96
3          95
4          95
5          93
6          92
7          92
8          92
9          92
```

```
query4 = """
SELECT SUBSTR(release_date, 1, 4) AS release_year, COUNT(*) AS count
FROM spotify
GROUP BY release_year
ORDER BY release_year;
"""
df4 = pd.read_sql_query(query4, conn)
print(df4)
```

```
   release_year  count
0          1959      1
1          1964      2
2          1965      3
3          1973      2
4          1976      2
5          1979      1
6          1981      1
7          1983      1
8          1986      1
9          1987      3
10         1988      1
11         1989      2
12         1991      1
13         1992      1
14         1993      3
15         1994      2
16         1996      1
17         1997      2
18         1999      1
19         2000      2
20         2001      4
21         2002      2
22         2004      1
23         2005      1
24         2006      6
25         2007      4
26         2008      4
27         2009      3
```

```
28          2010      3
29          2011      3
30          2012      2
31          2013      4
32          2014      5
33          2015      7
34          2016      3
35          2017     14
36          2018     21
37          2019     27
38          2020     14
39          2021     20
40          2022     22
41          2023     39
42          2024     65
43          2025      3
```

```
query5 = """
SELECT SUBSTR(release_date, 1, 4) AS release_year, AVG(popularity) AS
average_popularity
FROM spotify
GROUP BY release_year
ORDER BY release_year;
"""
df5 = pd.read_sql_query(query5, conn)
print(df5)
```

```
   release_year  average_popularity
0          1959           71.000000
1          1964           69.000000
2          1965           75.000000
3          1973           68.000000
4          1976           75.500000
5          1979           85.000000
6          1981           69.000000
7          1983           83.000000
8          1986           66.000000
9          1987           66.000000
10         1988           57.000000
11         1989           77.000000
12         1991           72.000000
13         1992           64.000000
14         1993           72.000000
15         1994           73.500000
16         1996           49.000000
17         1997           81.500000
18         1999           47.000000
19         2000           65.000000
20         2001           56.750000
21         2002           57.500000
```

```
22          2004           80.000000
23          2005           80.000000
24          2006           75.666667
25          2007           78.000000
26          2008           58.750000
27          2009           52.666667
28          2010           80.333333
29          2011           77.666667
30          2012           77.000000
31          2013           70.250000
32          2014           78.800000
33          2015           68.428571
34          2016           87.000000
35          2017           75.428571
36          2018           65.476190
37          2019           68.518519
38          2020           58.357143
39          2021           61.450000
40          2022           68.500000
41          2023           67.358974
42          2024           61.292308
43          2025           63.666667
```

```
query6 = """
SELECT artist_genre, COUNT(*) AS total_tracks
FROM spotify
WHERE artist_genre IS NOT NULL
GROUP BY artist_genre
ORDER BY total_tracks DESC;
"""
df6 = pd.read_sql_query(query6, conn)
print(df6.head())
```

```
                 artist_genre  total_tracks
0                 bedroom pop            17
1                       indie             7
2                         r&b             6
3                   dream pop             6
4  art rock, alternative rock             4
```