```
#Victor de Lima Souza
#199335
```

```
import pandas as pd
import sqlite3

def load_to_sql_in_chunks(filepath, tablename, conn, chunksize=100000):
    """
    Lê um arquivo TSV.GZ em blocos e grava no banco SQLite.
    """
    print(f"🔄 Carregando {tablename} ...")
    chunks = pd.read_csv(filepath, sep='\t', compression='gzip', na_values='\\N', chunksize=chunksize)
    for i, chunk in enumerate(chunks):
        chunk.to_sql(tablename, conn, if_exists='append', index=False)
        print(f"   Inserido chunk {i+1}")
```

```
conn = sqlite3.connect("imdb.db")
```

```
load_to_sql_in_chunks("title.basics0.tsv.gz", "basics", conn)
load_to_sql_in_chunks("title.ratings.tsv.gz", "ratings", conn)
load_to_sql_in_chunks("title.principals0.tsv.gz", "principals", conn)
```

```
   Inserido chunk 827
   Inserido chunk 828
   Inserido chunk 829
   Inserido chunk 830
   Inserido chunk 831
   Inserido chunk 832
   Inserido chunk 833
   Inserido chunk 834
   Inserido chunk 835
   Inserido chunk 836
   Inserido chunk 837
   Inserido chunk 838
   Inserido chunk 839
   Inserido chunk 840
   Inserido chunk 841
   Inserido chunk 842
   Inserido chunk 843
   Inserido chunk 844
   Inserido chunk 845
   Inserido chunk 846
   Inserido chunk 847
   Inserido chunk 848
   Inserido chunk 849
   Inserido chunk 850
   Inserido chunk 851
   Inserido chunk 852
   Inserido chunk 853
   Inserido chunk 854
   Inserido chunk 855
   Inserido chunk 856
   Inserido chunk 857
   Inserido chunk 858
   Inserido chunk 859
   Inserido chunk 860
   Inserido chunk 861
   Inserido chunk 862
   Inserido chunk 863
   Inserido chunk 864
   Inserido chunk 865
   Inserido chunk 866
   Inserido chunk 867
   Inserido chunk 868
   Inserido chunk 869
   Inserido chunk 870
   Inserido chunk 871
   Inserido chunk 872
   Inserido chunk 873
   Inserido chunk 874
   Inserido chunk 875
   Inserido chunk 876
   Inserido chunk 877
   Inserido chunk 878
   Inserido chunk 879
   Inserido chunk 880
   Inserido chunk 881
   Inserido chunk 882
   Inserido chunk 883
   Inserido chunk 884
```

```
# Quais são os 5 filmes com as maiores notas?
query1 = """
SELECT b.primaryTitle, r.averageRating, r.numVotes
```

```
FROM ratings r
JOIN basics b ON r.tconst = b.tconst
WHERE b.titleType = 'movie'
ORDER BY r.averageRating DESC, r.numVotes DESC
LIMIT 5;
"""
top5 = pd.read_sql_query(query1, conn)
display(top5)
```

	primaryTitle	averageRating	numVotes
0	Kaveri	10.0	1023
1	Kurukku	10.0	451
2	Jedal Dar Omghe 30 Metri	10.0	142
3	Sargashte	10.0	134
4	Gorgeous Rascal	10.0	115

Próximas etapas: Gerar código com top5 New interactive sheet

```
# Qual é o gênero mais frequente entre os filmes com nota maior que 8?
query2 = """
SELECT b.genres, COUNT(*) AS freq
FROM ratings r
JOIN basics b ON r.tconst = b.tconst
WHERE r.averageRating > 8 AND b.titleType = 'movie' AND b.genres IS NOT NULL
GROUP BY b.genres
ORDER BY freq DESC
LIMIT 1;
"""
genre_top = pd.read_sql_query(query2, conn)
display(genre_top)
```

	genres	freq
0	Documentary	7206

```
# Quais são os 3 atores/atrizes que mais participaram de filmes com nota maior que 7.5?
query3 = """
SELECT p.nconst, COUNT(*) AS qtd_filmes
FROM principals p
JOIN ratings r ON p.tconst = r.tconst
JOIN basics b ON b.tconst = p.tconst
WHERE r.averageRating > 7.5 AND b.titleType = 'movie' AND p.category IN ('actor', 'actress')
GROUP BY p.nconst
ORDER BY qtd_filmes DESC
LIMIT 3;
"""
top_actors = pd.read_sql_query(query3, conn)
display(top_actors)
```

	nconst	qtd_filmes
0	nm0004660	231
1	nm0595934	155
2	nm3183374	124

Próximas etapas: Gerar código com top_actors New interactive sheet

```
conn.close()
```