Python:

1. I first imported the necessary packages.
2. Then I began the ETL (Extract, Transform, Load) process, I established a connection to a MySQL database, where I first dropped and then created the base I used, ‘films’ This ensured I was working with a fresh database.
3. I extracted data from three sources MongoDB, an API call, and a CSV file
    a. I connected to a MongoDB instance using credentials and fetched movie data. The connection string consisted of the MongoDB cluster details, including the username and password. I extracted documents containing movie information.
    b. I then connected to the `trakt.tv` database via an API call to extract popular movie data. I decided to fetch data from multiple pages to get more than just the initial 10 data points the API provided by default.
    c. Lastly, I extracted data from an online CSV file containing movie information. This was achieved by simply reading the data into a Pandas DataFrame using `pd.read_csv()`.
4. I transformed the data so that it would be easy to utilize in MySQL
    a. I transformed the MongoDB data by selecting relevant fields like `title`, `year`, `directors`, and `ratings`. I also cleaned the data by dropping any missing values and ensured each field was of the correct data type.
    b. For the API data, I first normalized the data since it was in JSON format. Then, I selected the relevant columns and transformed them into a DataFrame. As with the MongoDB data, I cleaned the API data by setting appropriate data types and handling missing values.
    c. The CSV data was already loaded into a DataFrame, so there wasn't much transformation needed, however, I selected relevant variables, dropped any missing values, and renamed titles to follow the format I had used previously.
5. I then loaded the cleaned and transformed data into the MySQL 'films' database.

MySQL:

I designed a star schema in a MySQL database called ‘films’ to manage movie-related data, including details about movies, ratings, and profit. I chose a star schema because it's efficient for aggregations, and it works well with the ETL process implemented in Python.

1. I created and inserted data into three Dimension Tables:
    a. DimMovie, which captures movie attributes. After defining the structure of the table, I populate it with data using a JOIN operation on other tables, namely popular, profit, and ratings
    b. DimRatings, which captures ratings from different platforms. I add data to the table by selecting the relevant ratings from the ratings table.
    c. DimProfit, which provides financial data for the movies. I filled this table with data from the profit table.
2. I then created a central Fact Table:

    a. FactMovieMetrics connects all the dimensions. While this table doesn't contain direct, measurable metrics, it provides links to retrieve comprehensive insights about movies from the other tables. I then populated this fact table using JOIN operations on the popular, ratings, and profit tables, as well as the dimensions we just created.
3. I then performed a series of data analytic calculations
    a. First, I was intrigued to see how different movie genres fared financially on a global scale. To find this out, I grouped movies by their genres and calculated the average worldwide gross earnings for each group. Through this, I could discern which genres typically pulled in the most money. Such insights are valuable; for instance, they can help film studios decide which genre might be most profitable for their next venture
    b. Then to view the movie industry's yearly performance. I decided to summarize each year in terms of average IMDb ratings, total spent on movie budgets, total worldwide gross, and net profit. By comparing these metrics year by year, I aimed to observe trends and patterns. Perhaps the industry saw peaks in profitability during certain years, or maybe there were periods where the quality of movies (as reflected by average ratings) soared or dipped.
    c. Beyond just financials, I was curious about how different genres were received by audiences in terms of ratings. So, I calculated average ratings (from both Rotten Tomatoes and IMDb) and juxtaposed these with the financial metrics for each genre.
    d. Finally, to help determine if there was a correlation between a director's films returning large profits and receiving high ratings. I grouped movies by directors and computed their average worldwide earnings and IMDb ratings.