Agenda

- Project Overview
- Data Acquisition, Enrichment, and Examination
- SQL Database
- Analysis Dataframes
- Exploratory data analysis (EDA)
- Key Findings and Insight
- Challenges and Learnings
- Future Work and Improvements
- Final Conclusions

Project Overview & Business Case:

Build an **app** allowing:

✔️ **Users** – Analyze actors, directors and movie genres, ratings, popularity…

✔️ **Film production and distribution companies** – Data driven decision making

Important variables

- **Average Ratings:** IMDb average ratings per movie
- **Popularity**: The amount of votes received per movie at IMDb
- **Genre**: Thriller, Action, Comedy…
- **Duration**: Movie duration (minutes)

Project Management

Kanban Table

Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu
		Data cleaned	EDA - Tableau	SQL	Streamlit		Presentation & Environment	Executable & GitHub	Readme	

Pre-work - Datasets

- Movies project
  - Data Science End to End Project
    - Movie recommends → https://developer.imdb.com/non-commercial-datasets/
    - Predict ratings → https://www.kaggle.com/datasets/asaniczka/tmdb-
    - Predict revenue → ww.kaggle.com/datasets/rounakbanik/the-movies-dataset
  - Data Analytics End to End Project
    - SQL - Tableau
      Budget/Revenue?
      Actors and Genre/rRatings
      Evolution of genres/ movies over time
      → https://query.wikidata.org/#SELECT%20DISTINCT%20%3Fitem%20%3Fit

In process

- project presentation

Next Steps

- GitHub update
- Readme file

Blocked

Abandoned

- CHECK REVENUE ETC
- app executable

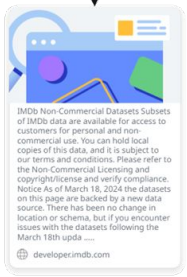
Completed

- Data selection & review
- Datacleaning: initial cleaning
- Data reviewing
- Data cleaning - outliers
- Data cleaning: final DataFrames
- Genres additional filter
- EDA quick review
- EDA - genres
- EDA - Directors
- EDA - Actors
- SQL Diagram sketch
- SQL Relational Data Model
- SQL Diagram
- SQL Database
- streamlit review - lesson on week 5
- streamlit sketch
- streamlit interactive plots
- environment

Rubric:

We want to produce a movie and we need to determine considering ratings and popularity:

- Genre
- Actor/s
- Director
- Duration? NO



Possibilities

- Python - SQL connection

Data Acquisition

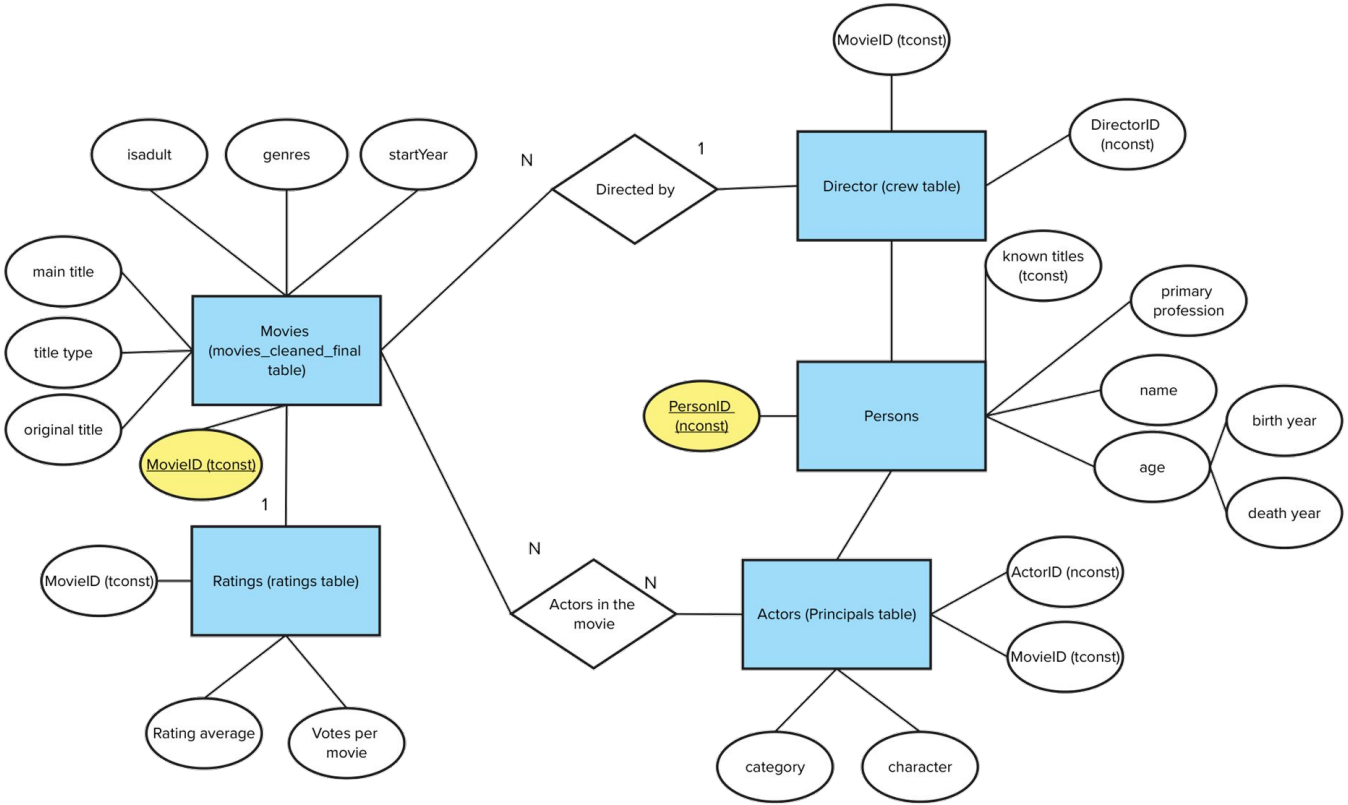
IMDb Non-Commercial Datasets:

- **Title.basics.tsv.g** → Movie and series titles basic information **(+11M rows)**
- **name.basics.tsv.gz** → Actors, writer, directors… **(+14M)**
- **title.ratings.tsv.gz** → Average ratings and vote counts per movie **(+1.5M)**
- **title.crew.tsv.gz** → Directors and writers per movie **(+11M)**
- **title.principals.tsv.gz** → Actors and characters per movie **(+91M)**

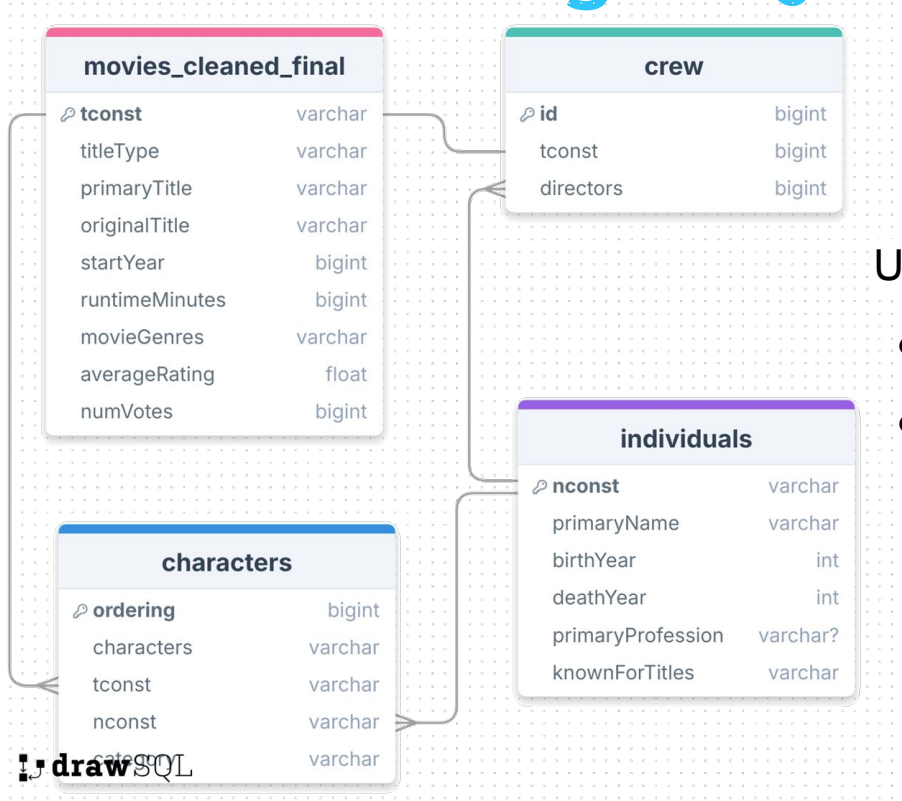
Kaggle:

- **Full TMDB Movies Dataset** → Ratings, votes, budget, revenue… **(1.2M)**

Database Design: Entity–relationship model



Database Design: SQL Diagram



Uploaded data from 2020 onwards:

- +47k movies
- + 7k persons

Analysis Dataframes

actors_df → +860k characters

✔️ Actors and actresses personal information
✔️ Movies and characters
✔️ Ratings and popularity

Movies_df → +98k movies

✔️ Release year, genre and duration
✔️ Directors
✔️ Ratings and popularity

Film production company

	Action	Drama	Comedy
Genre popularity	1st	2nd	3rd
Direction	Denis Villeneuve	John Krasinski	Adam Mckay
Cast	Scarlett Johansson	Leonardo DiCaprio	James Remar

Obstacles and solutions

🚧 **Loading huge Datasets into MySQL database causing significant delays**
- 🔹 **Solution:** Filtering data by year → 2010 onwards

🚧 **Not having consistent data for budget and revenue**
- 🔹 **Solution:** Abandoning budget-revenue approach

🚧 **Movies with few votes having weird average ratings**
- 🔹 **Solution:** Considering movies with high vote counts

🚧 **Huge characters dataset: +91M rows**
- 🔹 **Solution:** Adding serials/movies information to the dataset allowed filtering

Final Project Learnings

- Streamlit deployment
- Environment management
- Python - SQL connection
- Tableau - EDA performance