Turning IMDB Data into a Data Warehouse

Rafe Burns, Tony DiRubbo, Dylan Phillipps

Contents

Project Charter and Plan

The goal of this project is to build an analytical data warehouse using publicly available IMDB datasets. The team is assembling a complete project charter, outlining a full data warehouse architecture supported by dimensional models, fact and dimension tables, and bus matrices. A Python based ETL pipeline will load curated data into SQLite, and the resulting structures will support a set of functional requirements that focus on profiling, organizing, and preparing IMDB data for analysis. The project plan also includes the development of business intelligence dashboards in Power BI that reflect the analytical needs identified during data profiling.

The primary business processes modeled in this project center on analyzing movie ratings, tracking title releases, and monitoring long term trends within genres. IMDB contains extensive information about movies, television shows, cast and crew members, and audience ratings, making it valuable for insights into which movies perform best, how genres rise or decline, and how film production evolves over time. These outputs support real world decision making for streaming platforms and production studios seeking to guide content planning and advertising strategies. Within the team, Rafe will focus on planning and preparation, Tony will develop and present the ETL pipeline, and Dylan will create and present the business intelligence dashboards in Power BI.

IMDB Overview

IMDB has public data that individuals can use for non-commercial use
Source Tables Provided:

File	Purpose
`title.basics.tsv`	Title information (name, type, year, genres, runtime)
`title.ratings.tsv`	Average rating + number of votes
`name.basics.tsv`	People (name, birth year, profession)
`title.crew.tsv`	Director & writer IDs per title
`title.principals.tsv`	Cast and crew by title (with role)

Fact Tables and Bus Matrix

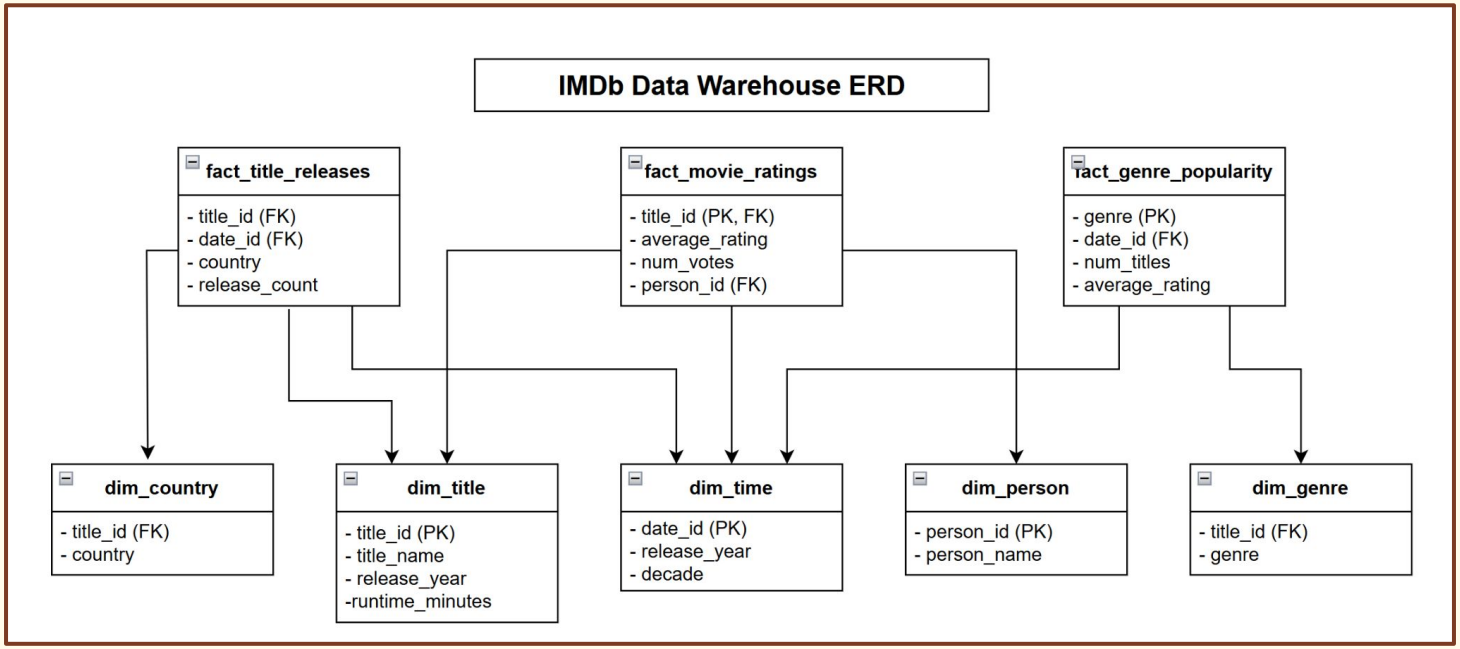
3 Main Business Processes:

1. Movie Ratings - How users rate movies and shows
2. Title Releases - How many titles are released over time
3. Genre Popularity - Which genres are most common or highest-rated

Bus Matrix:

Business Process Name	Fact Table	Fact Grain Type	Granularity	Facts	dim_title	dim_genre	dim_time	dim_person	dim_country
Movie Ratings	fact_movie_ratings	Transaction	One row per title with rating	average_rating, num_votes	X	X	X	X	
Title Releases	fact_title_releases	Transaction	One row per title released	release_count	X	X	X		X
Genre Popularity	fact_genre_popularity	Periodic Snapshot	One row per title per genre per year	num_titles, average_rating	X	X	X		

Data Warehouse Entity Relationship Diagram



Business Process 1: fact_movie_ratings

Column Name	Description	Data Type	PK	unique	not_null	FK Ref Dimension.col	Source
title_id	IMDb title identifier	TEXT	✅	✅	✅	dim_title.title_id	`title.ratings.tsv.gz`
average_rating	IMDb average rating	REAL			✅		`title.ratings.tsv.gz`
num_votes	Number of IMDb votes	INT			✅		`title.ratings.tsv.gz`
person_id	Director ID (optional link)	TEXT				dim_person.person_id	`title.crew.tsv.gz` + `name.basics.tsv.gz`

Dimension: dim_title

Column Name	Description	Data Type	PK	unique	not_null	Source
title_id	IMDb title identifier (e.g., tt0111161)	TEXT	✅	✅	✅	title.basics.tsv.gz
title_name	Movie’s primary title	TEXT			✅	title.basics.tsv.gz
release_year	Year the movie was released	INT				title.basics.tsv.gz
runtime_minutes	Movie runtime in minutes	REAL				title.basics.tsv.gz

Dimension: dim_person

Column Name	Description	Data Type	PK	unique	not_null	Source
person_id	IMDb person identifier (e.g., nm0000233)	TEXT	✅	✅	✅	`name.basics.tsv.gz`
person_name	Director’s full name	TEXT			✅	`name.basics.tsv.gz` joined via `title.crew.tsv.gz`

Business Process #2: fact_title_releases

Column Name	Description	Data Type	PK	unique	not_null	FK Ref Dimension.col	Source
title_id	IMDb title identifier	TEXT			✅	dim_title.title_id	`title.basics.tsv.gz`
date_id	Year of release (time dimension key)	INT			✅	dim_time.date_id	Derived from `release_year`
country	Release country or region	TEXT			✅	dim_country.country	`title.akas.tsv.gz`
release_count	Always 1 per movie release	INT			✅		Derived during ETL

Dimension: dim_time

Column Name	Description	Data Type	PK	unique	not_null	Source
date_id	Surrogate key (year, same as release_year)	INT	✅	✅	✅	Derived from `release_year`
release_year	Movie release year	INT			✅	Derived
decade	Decade grouping (e.g., 1990, 2000)	INT			✅	Derived

Dimension: dim_country

Column Name	Description	Data Type	PK	unique	not_null	Source
title_id	IMDb title identifier	TEXT			✅	`title.akas.tsv.gz`
country	Country or region code (e.g., US, GB)	TEXT			✅	`title.akas.tsv.gz` (region field)

Business Process #3: fact_genre_popularity

Column Name	Description	DataType	PK	unique	not_null	FK Ref Dimension.col	Source
genre	Movie genre	TEXT			✅	dim_genre.genre	Derived from `title.basics.tsv.gz`
date_id	Release year	INT			✅	dim_time.date_id	Derived
num_titles	Count of titles in that genre/year	INT			✅		Aggregated from `dim_genre` + `dim_title`
average_rating	Average rating for genre/year	REAL			✅		Aggregated from `fact_movie_ratings`

Dimension: dim_genre

Column Name	Description	Data Type	PK	unique	not_null	Source
title_id	IMDb title identifier (foreign key to `dim_title`)	TEXT			✅	`title.basics.tsv.gz`
genre	Movie genre (e.g., Action, Drama)	TEXT			✅	`title.basics.tsv.gz`

Data Warehouse Implementation

The data warehouse implementation follows the conventions and standards established in the project plan and uses the techniques introduced in the course, including the use of a staging area, conformed dimensions, and an enterprise bus structure. A Python based ETL pipeline was developed to extract the IMDB files, load them into a staging area with Pandas, and prepare them for transformation. During this process the data was filtered to include movies only, multivalued genres were normalized, and additional attributes such as decade and date_id were derived to support analytics. Conformed dimensions were created so that shared structures could be reused across all fact tables, and IMDB identifiers such as title_id and person_id were preserved as natural keys.

After transformations were completed, the ETL workflow loaded all dimension tables first and then populated the fact tables according to the defined grain. The fact_movie_ratings table contains one row per movie. The fact_title_releases table contains one row per title. The fact_genre_popularity table contains one row for each combination of genre and year. This layered and structured approach ensures that the data warehouse supports consistent analytics and aligns with the best practices emphasized throughout the course.

ETL Pipeline

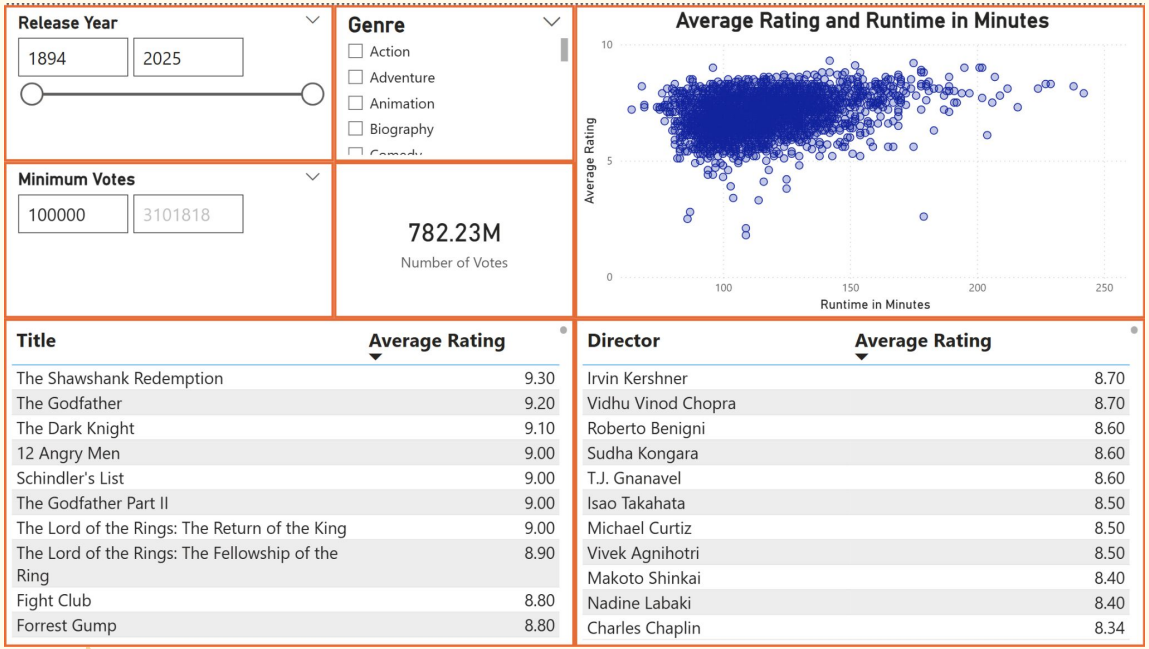
Consolidated into a Simple Python Script

```
18  # Directory for downloads
19  DOWNLOAD_DIR = "./imdb_downloads"
20  os.makedirs(DOWNLOAD_DIR, exist_ok=True)
21
22
23  # Download a dataset file if not already present
24  def download_imdb_file(filename):
25      url = IMDB_BASE_URL + filename
26      local_path = os.path.join(DOWNLOAD_DIR, filename)
27
28      if not os.path.exists(local_path):
29          print(f"Downloading {filename} ...")
30          response = requests.get(url, stream=True)
31          response.raise_for_status()
32
33          with open(local_path, "wb") as f:
34              for chunk in response.iter_content(chunk_size=8192):
35                  f.write(chunk)
36
```

ETL Pipeline

- Transforms source fields into dimensional structures such as title, genre, time, person, and country.
- Normalizes multivalued genre attributes into one row per title and genre.
- Merges ratings and computes derived attributes including decade and date_id.
- Preserves IMDb natural keys such as title_id and person_id across all dimensions and facts.
- Ensures ordered loading by inserting all dimension tables first, followed by fact tables.
- Supports ongoing updates to ratings, votes, genres, directors, and regional release data.
- UPSERT strategy acts as a survivorship rule by keeping the most recent data for each key.
- Data quality steps include removing nulls, enforcing numeric conversions, and validating keys before loads.
- Pipeline is safe to run daily, weekly, or on demand, ensuring consistent refresh of the warehouse.

BI Dashboard for Business Process #1

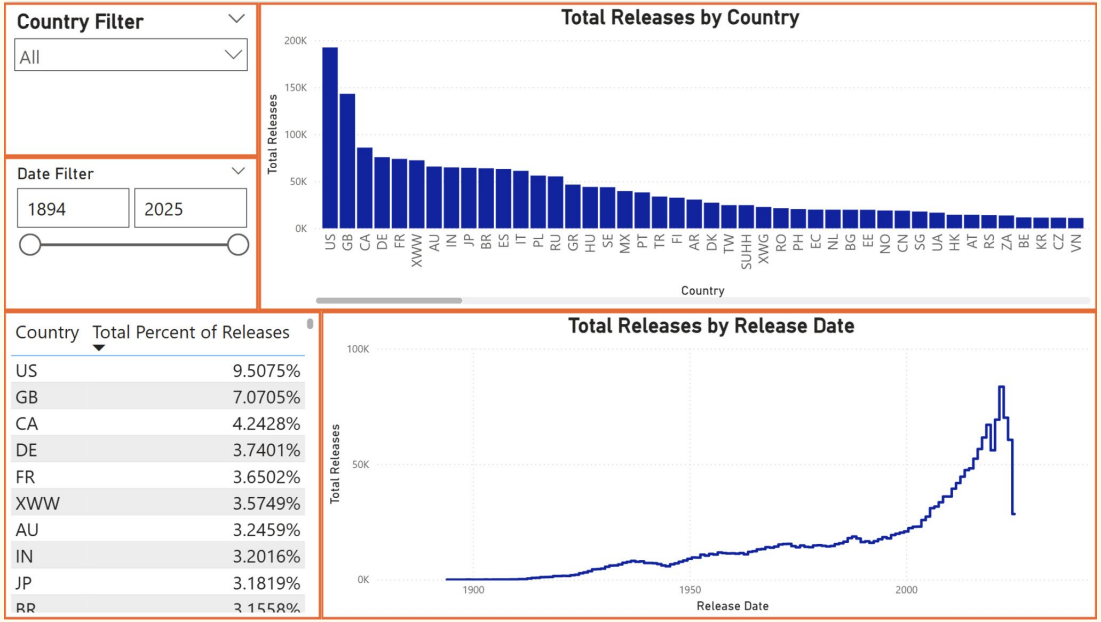


Title	Average Rating
The Shawshank Redemption	9.30
The Godfather	9.20
The Dark Knight	9.10
12 Angry Men	9.00
Schindler's List	9.00
The Godfather Part II	9.00
The Lord of the Rings: The Return of the King	9.00
The Lord of the Rings: The Fellowship of the Ring	8.90
Fight Club	8.80
Forrest Gump	8.80

Director	Average Rating
Irvin Kershner	8.70
Vidhu Vinod Chopra	8.70
Roberto Benigni	8.60
Sudha Kongara	8.60
T.J. Gnanavel	8.60
Isao Takahata	8.50
Michael Curtiz	8.50
Vivek Agnihotri	8.50
Makoto Shinkai	8.40
Nadine Labaki	8.40
Charles Chaplin	8.34

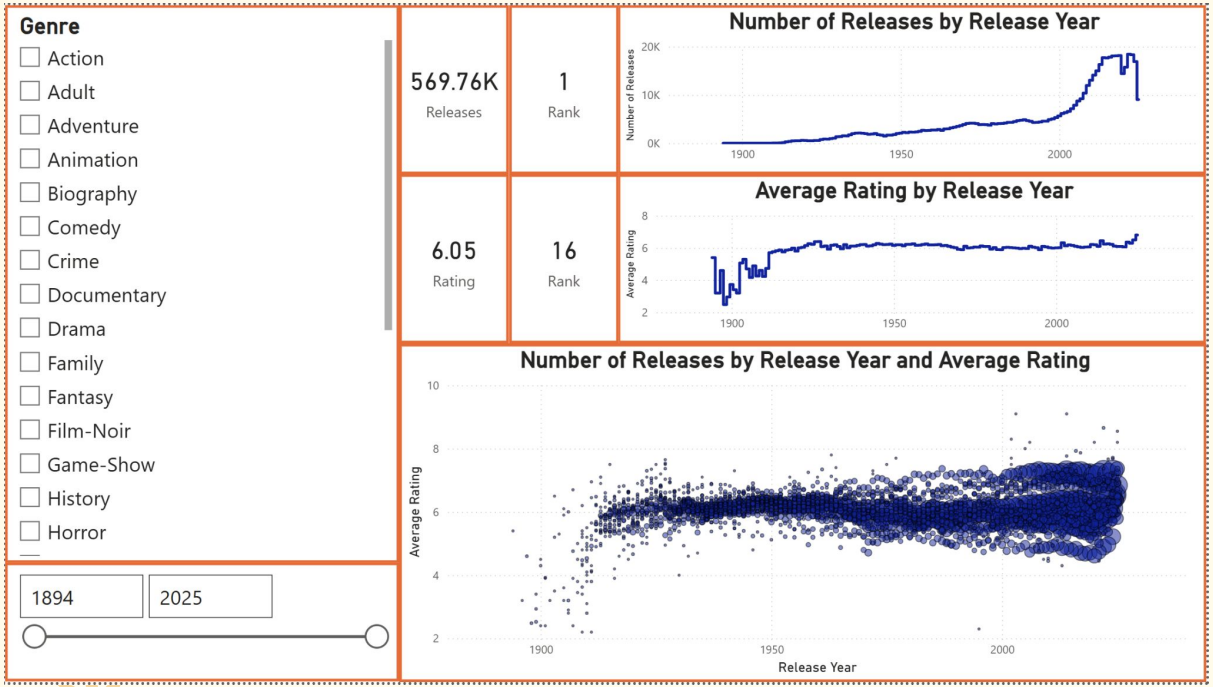
- Users filter by release year, genre, and minimum number of votes
- Displays relationship of rating and runtime
- Charts of top movies and directors

BI Dashboard for Business Process #2

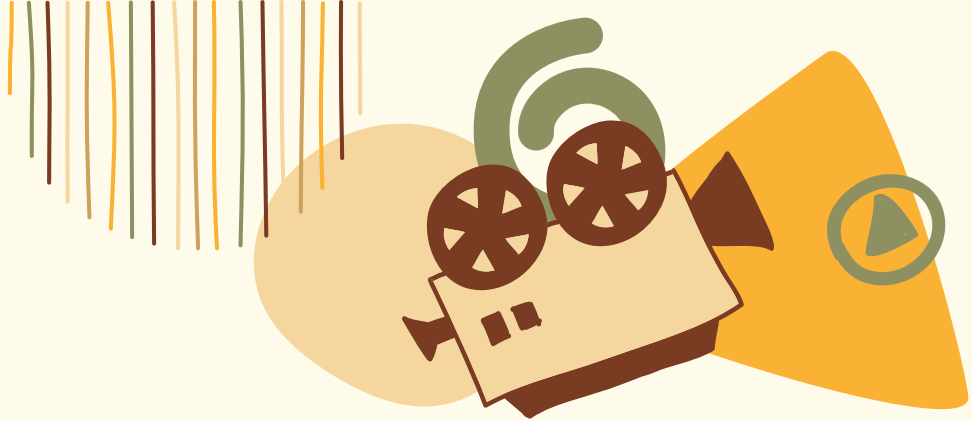


- Users filter by release year and country
- Graphs show number of releases by country and total releases over time
- Chart of percent share of total releases

BI Dashboard for Business Process #3



- Users filter by genre and release year
- Cards show releases and ratings in time frame, along with rank
- Bottom plot shows release year, rating, and volume



Any Questions?