Name: Fuad Choudhury

CIS4400, HOMEWORK-2, SPRING 2025

Professor: Jefferson Bien-Aime

Data Sourcing:

IMDb movie rating data selected for this project because it allows us to analyze trends and identify top-rated movies. The dataset was cleaned and uploaded to Azure Blob Storage. Since the raw dataset did not have a data dictionary, the data dictionary was created manually.

Data Sourcing Method:
Connection to Data Store (Azure Blob Storage)

Data Dictionary (Screenshots; Already uploaded on GitHub Repo):

`fact_movie_ratings:`

Column Name	Data Type	Description
`rating_id`	INT (PK)	Unique identifier for each rating record
`movie_id`	INT (FK)	Foreign key referencing `dim_movie.movie_id`
`date_id`	INT (FK)	Foreign key referencing `dim_date.date_id`
`average_rating`	FLOAT	Average IMDb rating of the movie
`num_votes`	INT	Number of user votes received by the movie

dim_movie:

Column Name	Data Type	Description
movie_id	INT (PK)	Unique internal identifier for the movie
tconst	VARCHAR	IMDb title identifier (e.g., "tt1234567")
primary_title	VARCHAR	Main movie title displayed to the user
original_title	VARCHAR	Original or native title of the movie
title_type	VARCHAR	Type of title (e.g., "movie", "short", "tvSeries")
is_adult	INT	Indicates if the movie is adult content (1 = Yes, 0 = No)
runtime_minutes	INT	Duration of the movie in minutes
genre	VARCHAR	Genre or comma-separated list of genres (e.g., "Drama,War")

dim_date:

Column Name	Data Type	Description
date_id	INT (PK)	Unique identifier for the date (year)
year	INT	Release year of the movie
decade	VARCHAR	Decade of the movie release (e.g., "1990s")

Storage:

The cleaned datasets were stored in Azure Blob Storage in an organized manner. Each dataset was uploaded as a separate CSV file: dim_movie.csv, dim_date.csv, and fact_movie_ratings.csv.

Storage of Choice:
Azure Blob Storage

Scripts:
Python scripts were written to clean and upload the data, and Snowflake was used to create tables and load the data using SAS tokens/manually from the device.

Transformation:

The data transformation steps below:

- Converting all date formats to YYYY-MM-DD.
- Creating a separate dim_date table with columns for year and decade.
- Removing rows with null averageRating.
- Creating a surrogate key in the fact table.
- Ensuring correct datatypes and value ranges.
- Generating a data mapping to track transformations.

Data Mapping Tables:

Source File	Source Column	Transformed Field	Data Type	Description	Target Table
fact_movie_ratings.csv	rating_id	rating_id	INTEGER	Surrogate key for fact table	fact_movie_ratings
fact_movie_ratings.csv	movie_id	movie_id	INTEGER	FK to dim_movie	fact_movie_ratings
fact_movie_ratings.csv	date_id	date_id	DATE	FK to dim_date	fact_movie_ratings
fact_movie_ratings.csv	averageRating	averageRating	FLOAT	Average IMDb rating	fact_movie_ratings
fact_movie_ratings.csv	numVotes	numVotes	INTEGER	Number of users voted	fact_movie_ratings
dim_movie.csv	movie_id	movie_id	INTEGER	Primary key	dim_movie
dim_movie.csv	original_title	original_title	VARCHAR	Title of the movie	dim_movie
dim_date.csv	full_date	full_date	DATE	The full date	dim_date

dim_date.csv	year	year	INTEGER	Year extracted from date	dim_date
dim_date.csv	decade	decade	VARCHAR	Decade for time-based aggregation	dim_date

---

Modeling

For this project a star schema with one fact table fact_movie_ratings and two-dimension tables dim_movie and dim_date. Foreign keys were created between the fact and dimension tables using surrogate keys.

Modeling Tool:
Using Draw.io (ER diagram)

Fact Table:
fact_movie_ratings (rating_id, movie_id, date_id, averageRating, numVotes)

Dimension Tables:

- dim_movie (movie_id, original_title, genres)

- dim_date (date_id, full_date, year, decade)

Data Warehouse:
Created in AWS Redshift.

Data Warehouse Creation:

- **SQL ran to CREATE TABLE scripts** in Snowflake SQL-style for:
  - dim_movie
  - dim_date
  - fact_movie_ratings
- **Populated tables** using cleaned CSVs (via COPY INTO or equivalent commands).
- **Verified the structure** using DESCRIBE TABLE to confirm column names and data types.
- **Used the dimensional model**: fact table (with surrogate key rating_id) and dimension tables (with keys like movie_id, date_id).
- **Connected it to Tableau** to visualize and serve data from the warehouse.

Serving Data:

It’s done by Tableau Public to the exported CSV datasets and created interactive visualizations.

Visualizations:

1. **Top 10 Movies by Average Rating** (Bar Chart with filter > 100 votes)

2. **Movie Ratings and Votes Over Year** (Line Chart with release year)

3. **Movie Ratings and Votes by Genre** (Column Chart)

Filter: Global filter added for year range.

Visualization:

Serving Tool: Tableau Public
Public Link: Tableau workbook uploaded on Brightspace assignment-2 submission with this word file.

A Screenshot of Tableau Dashboard: