EXTRACT

IMDB Non-Commercial Datasets
https://datasets.imdbws.com/

- name.basics.tsv (907 MB)
- ~~title.akas.tsv~~ (dropped)
- title.basics.tsv (1 GB)
- title.crew.tsv (394.4 MB)
- title.episode.tsv (241.3 MB)
- title.principals.tsv (4.2 GB)
- title.ratings.tsv (28.3 MB)

Total: 6.8 GB

The Oscar Award, 1927-2025 Dataset
https://www.kaggle.com/datasets/unanimad/the-oscar-award

- full_data.csv (2.3 MB)

MySQL Server (imdb_source Database)

Schema:
- title_basics
- title_crew
- title_episode
- title_principals
- title_ratings
- name_basics
- oscar_awards

TRANSFORM

[1] Look-up Table for Fixed-Length Values
- DimGenres and DimProfessions contain all values for genres and professions respectively as a look-up table.

[2a] Parsing Fixed-Length String Values
- Transform into one-hot encoded string
- Use WHILE loop to iterate over every value and get each value using SUBSTRING_INDEX
- genre = SUBSTRING_INDEX(SUBSTRING_INDEX(professions, ',', i), ',', -1));
- Assign true or false values to one-hot encoded string
- Ex.
  - genres_list = ['Action','Documentary','Fantasy']
  - input = 'Action,Fantasy'
  - output = 'TFT'
- Columns for one-hot encoding are added whenever necessary

[2b] Parsing Variable-Length String Values
- Use RECURSIVE CTEs
- WITH RECURSIVE split_nominees AS (
  Base Case
  UNION ALL
  Recursive Case
)
- nomineeIds = 'nm0000001,nm0000002,nm0000003'
- output:
  - row 1: person_key = 'nm0000001'
  - row 2: person_key = 'nm0000002'
  - row 3: person_key = 'nm0000003'
- Use cte_max_recursion_depth = 10000;

[3] Assigning Data Types
- num_votes = "10000" -> 10000 (INT)
- avg_rating = "6.7" -> 6.7 (FLOAT)
- end_year = "\N" -> null (nullable INT)
- full_name = null -> "unknown" (Assign default value)
- is_winner = "True" -> 1 (TINYINT/BOOLEAN)

[4] Assigning Primary and Foreign Keys
- Assign primary key to ID values
- Ex. title_key = 'tt0000001' from DimTitle and person_key = 'nm0000001' for DimPersons
- Foreign key constraints assigned to other tables for these values
- Fact tables have their primary keys auto incremented

[5] Data Validation
- Use INSERT IGNORE INTO to truncate rows that have either unclean data (eg. 'Reality-TV' in end_year column) or rows that don't follow foreign key constraints

LOAD

Process of Loading the Data from Source to the Data Warehouse

[1] Independent Dimension Tables

DimTitle & DimPerson
- Genres and Professions parsed
- Uses IDs from the datasets

[2] Bridge Dimension Tables

BridgeCrew
- Combines title_crew and title_principals
- Separates values by job, category, and character
- Assigns category based on whether person is a 'director' or 'writer' in title_crew

[3] Hierarchical Dimension Tables

DimEpisode
- Self-referencing from DimTitle
- Hierarchical relationship between series and episode

[4] Fact Tables

FactRatings
- Genre and Episode to reference for faster querying
- Success score computed

FactOscarAwards
- Uses the oscar_awards table
- Details the title and person of each nominated entity (if applicable)

FactCrewPerformancePerFilmGenre
- Similar to FactRatings but instead joins the crews to query the performance of persons faster