# Data Lineage Documentation

## Data Modification and Generation

### 1 Generating User Data

Script/Tool Used:

- Raw_data_generator.ipynb

Description:

Synthetic user data is generated in a Jupyter Notebook, including random nationality, gender, and age for 610 users.

Input Data:

- Number of users in the Movielens dataset (610)

Transformation/Logic:

- Randomized generation of user attributes.
- Data structured into a CSV format.

Output Data:

- File Name: users.csv
- Directory: movie_data/

### 2 Formatting Rating Data

Script/Tool Used:

- Raw_data_generator.ipynb
- Movielens dataset

Description:

The rating data from the Movielens dataset is reformatted into a JSON structure along with links to IMDB and TMDB datasets to increase project complexity.

Input Data:

- ratings.csv (Movielens dataset)
- movies.csv (Movielens dataset)
- links.csv (Movielens dataset)

Transformation/Logic:

Conversion of structured rating data from its original format into JSON.

Output Data:

- File Name: raw_unstructured_data.json
- Directory: movie_data/

# Data Pipeline Documentation

## Task 1: ingest_movielens

Function: ingest_movielens

Description:

Reads the raw_unstructured_data.json file from the movie_data directory.

Loads and returns the data in JSON format for further processing.

Input:

File: ../movie_data/raw_unstructured_data.json

Output:

JSON data (loaded into memory).

## Task 2: ingest_tmdb

Function: ingest_tmdb

Description:

Reads tmdb_5000_movies.csv and tmdb_5000_credits.csv files from the movie_data directory.

Returns the loaded data as Pandas DataFrames.

Input: TMDB dataset

Files:

../movie_data/tmdb_5000_movies.csv

../movie_data/tmdb_5000_credits.csv

Output:

Two DataFrames (movies and credits).

## Task 3: ingest_users

Function: ingest_users

Description:

Reads the users.csv file containing user demographic data.

Returns the loaded data as a Pandas DataFrame.

Input:

File: ../movie_data/users.csv

Output:

A DataFrame containing user data.

## Task 4: process_data

Function: process_data

Description:

Processes the ingested data by applying various transformations, including:

1. Preprocessing Movielens Ratings:
2. Extracts ratings and movie identifiers.
3. Merges ratings with TMDB IDs.
4. Preprocessing TMDB Movies:
5. Cleans and transforms movie attributes, such as genres, keywords, and production companies.
6. Adds new columns like age_restriction and high_budget.

Preprocessing Cast and Crew:

Extracts actor and director details from credits data.

Holiday Data Generation:

Adds holiday and seasonal indicators based on movie release dates.

Saves the processed datasets as CSV files in the movie_data directory.

Input:

DataFrames: Movielens ratings, TMDB movies, and credits.

Output:

Processed CSV files:

1. ratings.csv

2. tmdb.csv
3. cast.csv
4. crew.csv
5. Holidays.csv

Output Directory:

../movie_data

# Data Processing Details

### Preprocessing Movielens Data

1. Extracts movie IDs, IMDB IDs, TMDB IDs, and user ratings.
2. Cleans and converts numeric fields.
3. Merges ratings data with TMDB IDs.

### Preprocessing TMDB Movies

Extracts and cleans attributes like keywords, genres, production companies, and release dates.
Adds derived attributes:

- age_restriction: Based on genres and predefined rules.
- high_budget: Indicates whether the budget exceeds $1,000,000.

### Preprocessing Cast and Crew

Extracts actor information (name, character, gender) from cast data.
Extracts crew information (director details) from crew data.

### Holiday Data Generation

Flags movies released near specific holidays (e.g., Christmas, New Year).
Adds seasonal indicators (e.g., summer, spring).

# Data Flow and Lineage:

### Data Preprocessing:

Several preprocessing steps occur before loading data into the database. This includes:

1. CSV Data Reading: CSV files are read into Pandas DataFrames for processing.
2. Data Cleaning: Some columns, such as genres and keywords, are preprocessed to split values into multiple columns (e.g., genre1, genre2, genre3).
3. Unique Crew and Cast Processing: Crew and Cast information is cleaned and duplicate gender and name pairs are removed.

## Data Insertion

**User_dimension**: User data is loaded directly from users.csv into the User_dimension table.

**Date_dimension**: Data from holidays.csv is used to populate the Date_dimension table. The tmdbId is renamed to id.

**Genre_dimension**: Preprocessed data from the tmdb.csv file is loaded into Genre_dimension. Columns like genre1, genre2, genre3, and age_limit are derived from the genres and age_restriction columns in the source file.

**Keyword_dimension**: Similar to genre processing, the keywords column from tmdb.csv is split into three columns (keyword1, keyword2, keyword3) and loaded into Keyword_dimension.

**Crew_dimension**: Data from crew.csv is processed, with unique crew members (based on name and gender) being inserted into the Crew_dimension table.

**Cast_dimension**: Data from cast.csv is similarly processed, with unique cast members inserted into Cast_dimension.

**Search_Crew_bridge**: The crew data is used to create the Search_Crew_bridge table, linking crew members to their roles and departments in the movies.

**Search_Cast_bridge**: Similarly, cast data is used to create the Search_Cast_bridge, linking actors to their roles.

**Movie_dimension**: Data about movies (including title, language, and overview) from tmdb.csv is loaded into Movie_dimension.

**Search_fact Table**:

- This table consolidates data from multiple sources: ratings, movie details, crew, cast, genres, keywords, and release dates.
- The Search_fact table is populated with detailed information about user interactions with movies, linking each entry to the relevant dimension tables using foreign keys (e.g., user_ID, movie_ID, genre_ID, etc.).

- A unique ID is generated for each row in the fact table (search_fact_generate_id), and each fact row is inserted in bulk for efficiency.