Information Technology Institute

Track Data Engineering

Movies Production Insights Pipeline

- **Abdullah Mahmoud Ghayad**
- **Osama Tarek**
- **Omar Mahmoud**
- **Ataa Mohamed**
- **Youssef Ashraf**
- **Mohamed Selim**

Table Of Content

I- Abstract

The Movies Production Insights Pipeline Project aimed to design and implement an efficient data infrastructure for the movie industry. The primary objectives included:

1- Data Integration: Consolidate diverse data sources related to persons, movies, genres, production, and more into a centralized data warehouse.

2- Automation: Utilize Apache Airflow for automated orchestration of data pipelines, ensuring timely and accurate data processing.

3- Transformation: Employ Pyspark for effective data extraction and transformation, enabling the conversion of raw data into a structured format suitable for analysis.

4- Staging Area: Implement MongoDB as a NoSQL staging area for interim data storage, providing flexibility and scalability.

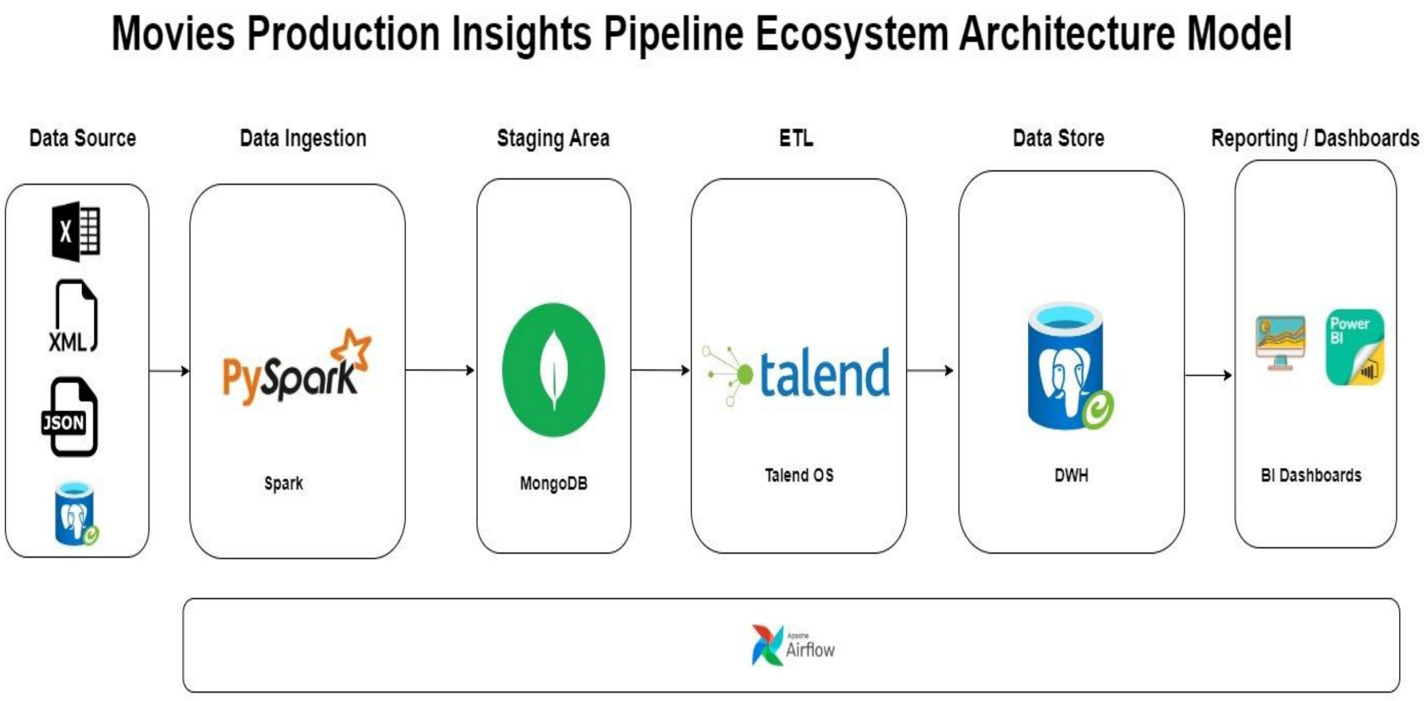
5- ETL Process: Utilize Talend for Extract, Transform, Load (ETL) processes, facilitating the seamless transfer of data from the staging area to the PostgreSQL data warehouse.

6- Change Data Capture (CDC): Implement a CDC strategy using the Meta_Data table to capture and track changes, ensuring the data warehouse reflects the latest information.

7- Visualization: Integrate Power BI for insightful visualization, allowing users to explore trends, patterns, and relationships within the movie industry.

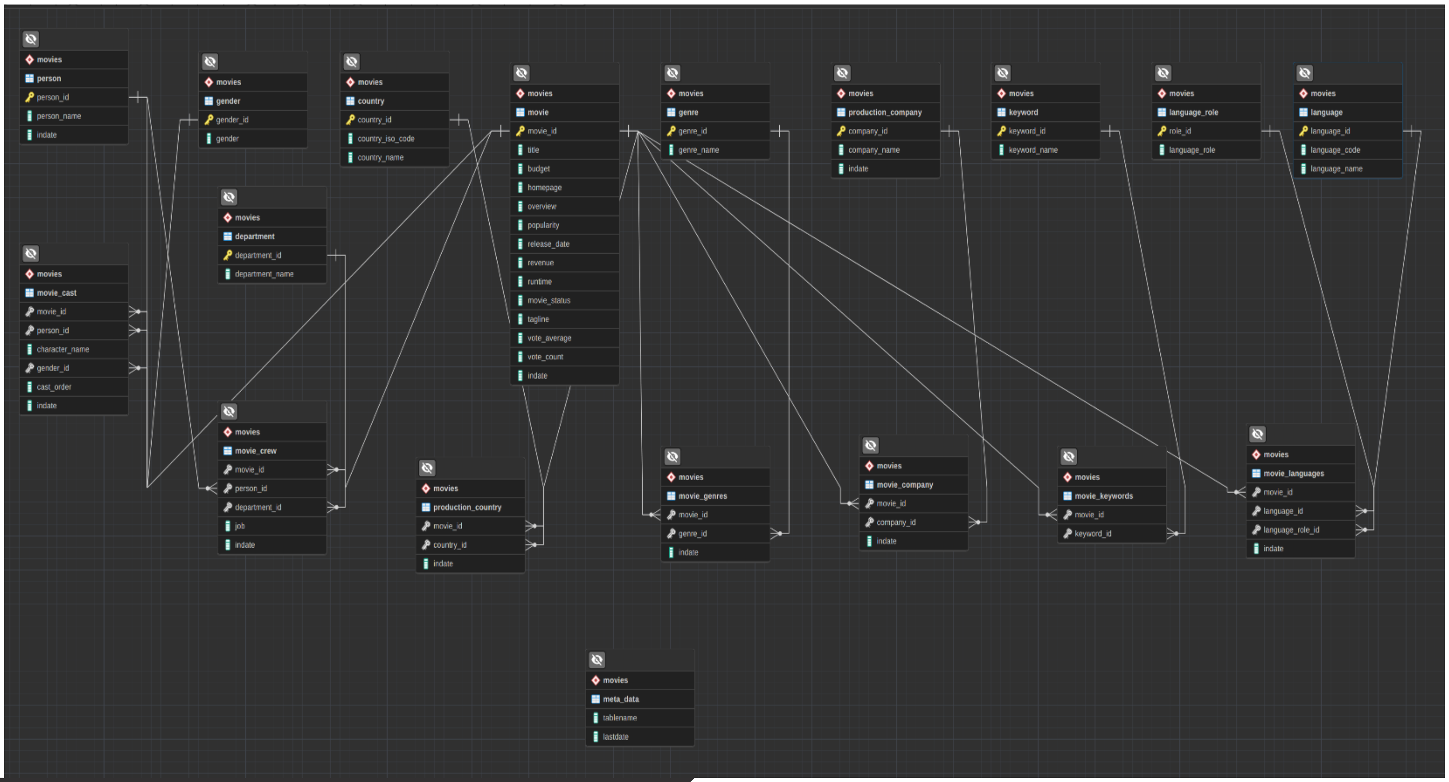
This documentation provides a high-level overview of the project's objectives, methodologies, and implementation strategies.

II- ecosystem solution



III- Data Sources

The Movies Production Insights Pipeline Project amalgamates data from diverse sources, creating a cohesive and extensive dataset. While the primary source comprises tables within a relational



database, a subset of tables, namely Country, Department, Genre, Language, and Language_Role, originates from external files in formats such as XML, JSON, and XLS.

Relational Database

The core of the dataset resides within the relational database, hosting tables that capture intricate details of the movie industry. Key tables include:

Person

- person_id: Unique identifier for individuals.
- person_name: Name of the person.
- indate: Date of data entry.

Movie_Cast

- movie_id: Unique identifier for movies.
- person_id: Identifier for persons involved in the cast.
- character_name: Character name played by the person.
- gender_id: Identifier for gender.
- cast_order: Order of appearance in the cast.
- indate: Date of data entry.

Movie

- movie_id: Unique identifier for movies.
- title: Title of the movie.
- budget: Budget allocated for the movie.
- homepage: Movie's homepage.
- release_date: Date of movie release.
- ... (Additional fields capturing various aspects of a movie)

Production_Company

- company_id: Unique identifier for production companies.
- company_name: Name of the production company.
- indate: Date of data entry.

Non-Relational Data

Country : Derived from external files, the Country table enriches the dataset with information about countries relevant to the movie industry:

- country_id: Unique identifier for countries.
- country_iso_code: ISO code for the country.
- country_name: Name of the country.

Department : The Department table provides details about different departments involved in movie production:

- department_id: Unique identifier for departments.
- department_name: Name of the department.

Genre : Information about movie genres is acquired from external files through the Genre table:

- genre_id: Unique identifier for genres.
- genre_name: Name of the genre.

Language

- External data sources contribute to the Language table:
- language_id: Unique identifier for languages.
- language_code: Code representing the language.
- language_name: Name of the language.

Language Role The Language_Role table complements language data by providing roles associated with each language:

- role_id: Unique identifier for language roles.
- language_role: Role associated with the language.

Keyword Exclusion

- Noteworthy is the strategic exclusion of the Keyword and Movie_Keywords tables from the staging area. Originally part of the relational database, their exclusion aligns with the project's focus on relevant and impactful data.
- This selective approach ensures a streamlined and optimized data processing pipeline, emphasizing the importance of data relevance and efficiency.

Meta_Data

- table_name: Name of the table.
- last_date: Date of the last update.

IV- Data Extraction

Files Data:

- Country Data
- Department Data
- Gender Data
- Genre Data
- Language Data
- Language Role Data

Steps

- **Step 1: Read Data**

  The function uses Apache Spark to read data from the specified path, creating a Spark DataFrame.

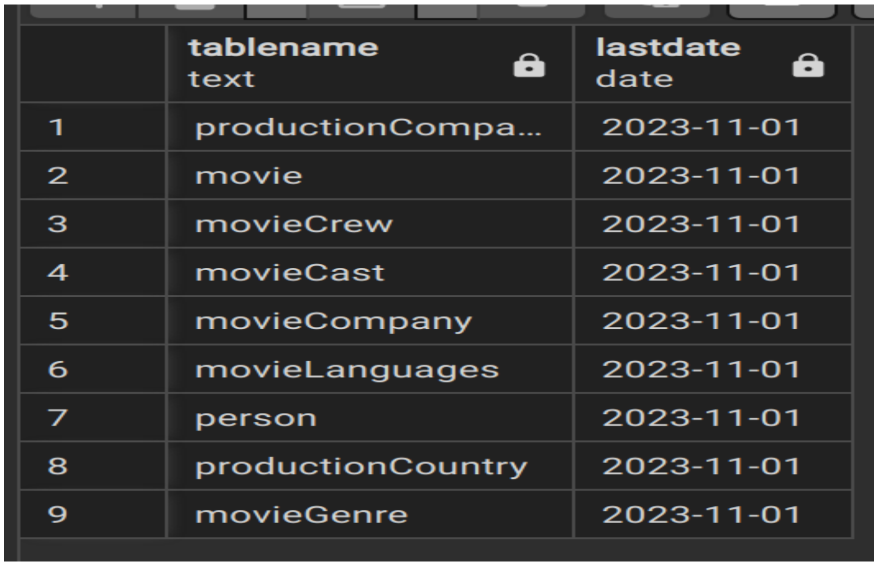
- **Step 2: Convert to Pandas DataFrame**
  The Spark DataFrame is then converted to a Pandas DataFrame. This facilitates compatibility with other data processing tools and libraries.
- **Step 3: Extract Filename and Archive**
  The function leverages the input_file_name() function to extract the filename of the processed file. It then uses the Archiving function to archive the file, ensuring that it is not processed again in subsequent DAG runs, preventing duplication.
- **Step 4: Return Result**
  The function returns the Pandas DataFrame containing the data, making it available for further processing within the data pipeline.
  These steps outline the general process followed by each function in extracting data from different file sources and ensuring data integrity through archiving.

The following functions are designed for data extraction from a PostgreSQL database utilizing Change Data Capture (CDC) principles. Each function performs a selective extraction based on the last update date and subsequently updates the metadata information in the meta_data table.

1. UpdateMetaData

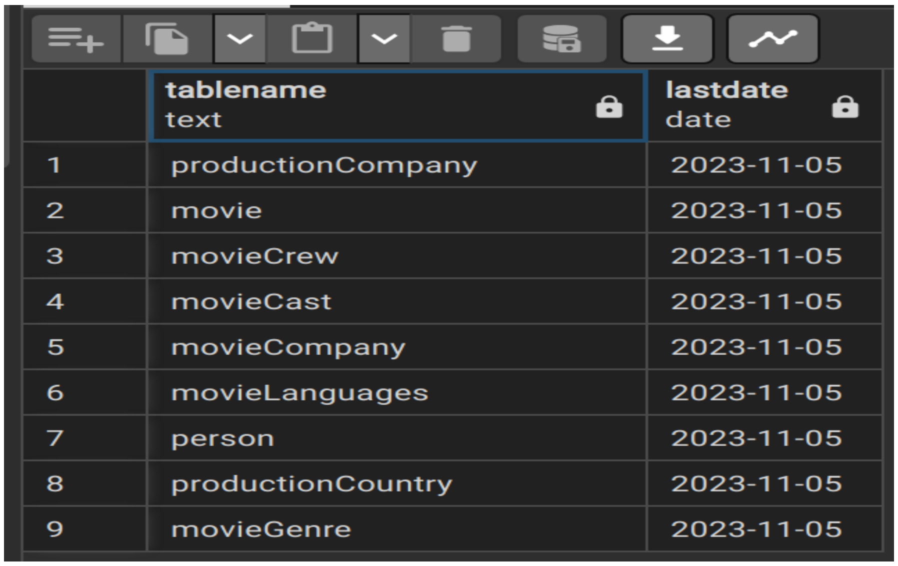
Purpose:Updates metadata information in the meta_data table, indicating the last update date for a specific table.

Before Update Meta Table :



	tablename text	lastdate date
1	productionCompa...	2023-11-01
2	movie	2023-11-01
3	movieCrew	2023-11-01
4	movieCast	2023-11-01
5	movieCompany	2023-11-01
6	movieLanguages	2023-11-01
7	person	2023-11-01
8	productionCountry	2023-11-01
9	movieGenre	2023-11-01

After Update Meta Table :



	tablename text	lastdate date
1	productionCompany	2023-11-05
2	movie	2023-11-05
3	movieCrew	2023-11-05
4	movieCast	2023-11-05
5	movieCompany	2023-11-05
6	movieLanguages	2023-11-05
7	person	2023-11-05
8	productionCountry	2023-11-05
9	movieGenre	2023-11-05

2. MoviePostgres

Purpose:Extracts data from the movies.movie table.

3. MovieGenrePostgres

Purpose:Extracts data from the movies.movie_genres table.

4. MovieCastPostgres

Purpose:Extracts data from the movies.movie_cast table.

5. MovieCompanyPostgres

Purpose:Extracts data from the movies.movie_company table.

6. MovieCrewPostgres

Purpose:Extracts data from the movies.movie_crew table.

7. MovieLanguagesPostgres

Purpose:Extracts data from the movies.movie_languages table.

8. PersonPostgres

Purpose:Extracts data from the movies.person table.

9. ProductionCompanyPostgres

Purpose:Extracts data from the movies.production_company table.

10. ProductionCountryPostgres

Purpose:Extracts data from the movies.production_country table.

Common Steps for Data Extraction Functions:

- Reads data from the specified table in the PostgreSQL database where the indate is greater than the last recorded date for the respective table.
- Converts the Spark DataFrame to a Pandas DataFrame for further processing.
- Retrieves the maximum indate to update the metadata using the UpdateMetaData function.
- Calls the UpdateMetaData function to update the last update date for the respective table.
- Returns the Pandas DataFrame containing the extracted data.
- These functions collectively implement a robust CDC process, ensuring efficient and incremental data extraction while maintaining metadata accuracy.

V- Data Staging (Loading to Stage Area)

- **Load Into Staging Area Function**

The LoadIntoStagingArea function is responsible for loading data into the MongoDB Staging Area for the Movies Production Insights Pipeline Project. This function takes various DataFrames as input and writes them to their respective collections in the MongoDB Staging Area.

Purpose:Transforms and loads data into the MongoDB Staging Area collections.

Input Parameters:

- country: DataFrame containing country data.
- department: DataFrame containing department data.
- gender: DataFrame containing gender data.
- genre: DataFrame containing genre data.
- language: DataFrame containing language data.
- language_role: DataFrame containing language role data.
- movie: DataFrame containing movie data.
- movieGenre: DataFrame containing movie genre data.
- movieCast: DataFrame containing movie cast data.
- movieCompany: DataFrame containing movie company data.
- movieCrew: DataFrame containing movie crew data.
- movieLanguages: DataFrame containing movie languages data.
- person: DataFrame containing person data.
- production_company: DataFrame containing production company data.
- production_country: DataFrame containing production country data.

Steps:

- Convert Pandas DataFrames to Spark DataFrames for compatibility with the MongoDB connector.
- Write each Spark DataFrame to its respective MongoDB collection in the Movies Staging Area.
- Utilize the MongoDB URI to specify the destination for each collection.
- Overwrite the existing data in MongoDB collections to ensure the Staging Area reflects the latest updates.

VI- Trigger Talend ETL Task

- The Trigger_Talend_ETL task serves as a pivotal component in the orchestration of the Movies Production Insights Pipeline.
- It is responsible for initiating the execution of the second DAG, Talend_ETL, once the first DAG successfully completes its tasks.

Task Overview:

- Task Identifier: Trigger_Talend_ETL
- Triggered DAG Identifier: Talend_ETL

Task Functionality:

Upon the successful completion of the initial DAG, this task functions as a trigger to seamlessly transition to the next phase of the pipeline.

Execution Flow:

First DAG Completion:

The task is configured to execute upon the successful completion of the first DAG.

This implies that all the necessary preliminary tasks, such as data extraction and staging, have been successfully executed.

Initiation of the Second DAG:

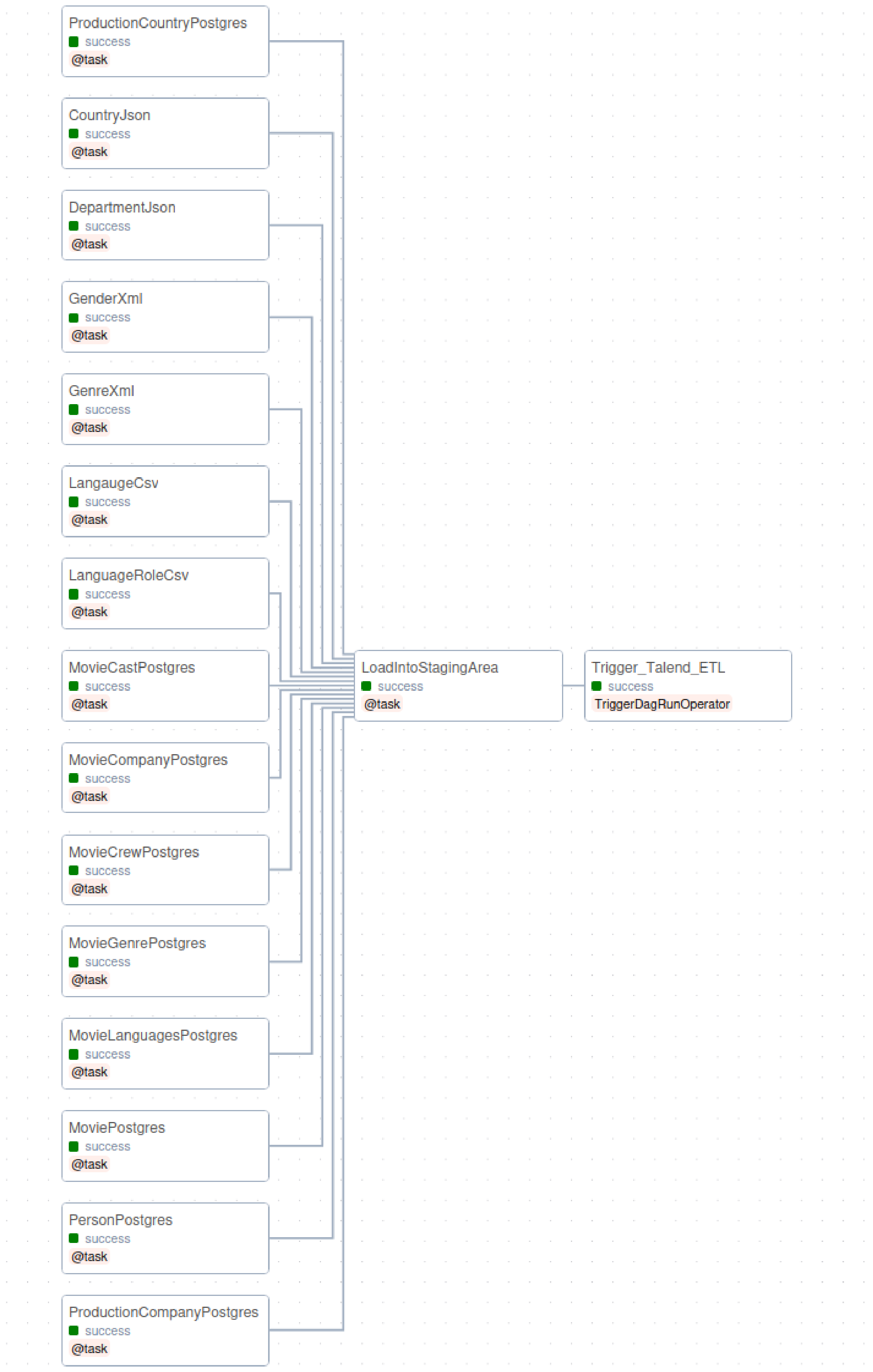
Once triggered, the task initiates the execution of the Talend_ETL DAG, signaling the beginning of the Extract, Transform, and Load (ETL) operations.

Data Warehouse ETL:

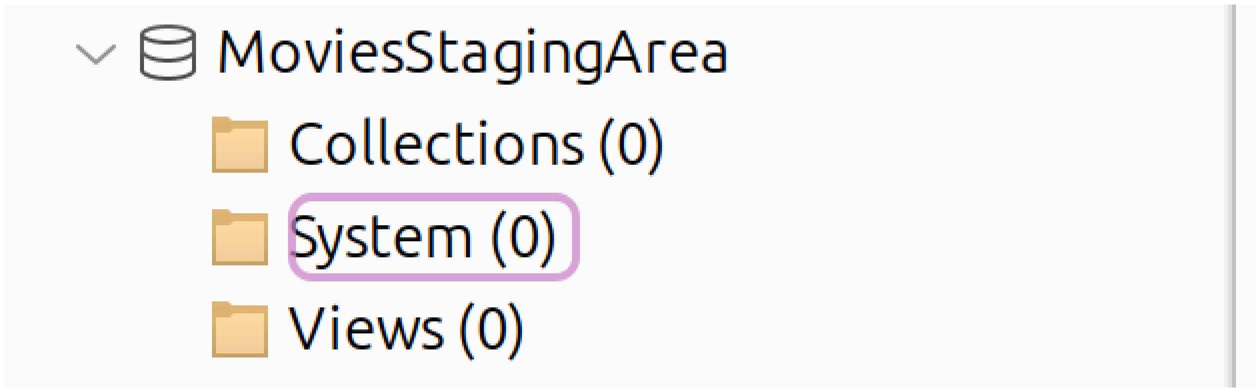
The Talend_ETL DAG is responsible for orchestrating ETL processes, moving data from the MongoDB Staging Area to the designated data warehouse.

Purpose: This task facilitates the seamless transition between different stages of the data pipeline. By triggering the Talend_ETL DAG, it ensures that the processed data is efficiently transferred to the data warehouse for further analysis and insights generation.

First DAG

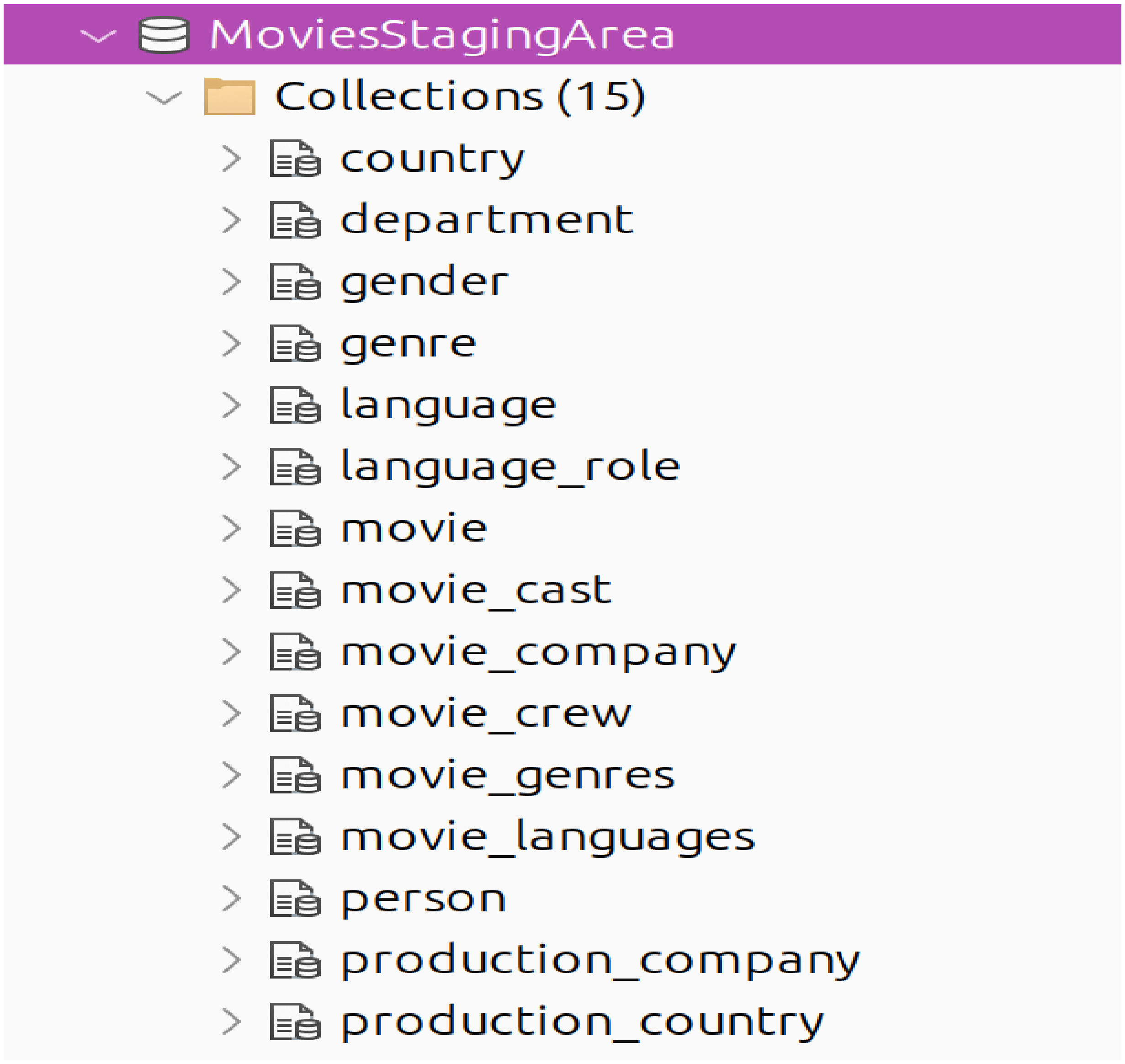

ProductionCountryPostgres
success
@task
CountryJson
success
@task
DepartmentJson
success
@task
GenderXml
success
@task
GenreXml
success
@task
LangaugeCsv
success
@task
LanguageRoleCsv
success
@task
MovieCastPostgres
success
@task
MovieCompanyPostgres
success
@task
MovieCrewPostgres
success
@task
MovieGenrePostgres
success
@task
MovieLanguagesPostgres
success
@task
MoviePostgres
success
@task
PersonPostgres
success
@task
ProductionCompanyPostgres
success
@task
LoadIntoStagingArea
success
@task
Trigger_Talend_ETL
success
TriggerDagRunOperator

Before Running Dag



MoviesStagingArea
Collections (0)
System (0)
Views (0)

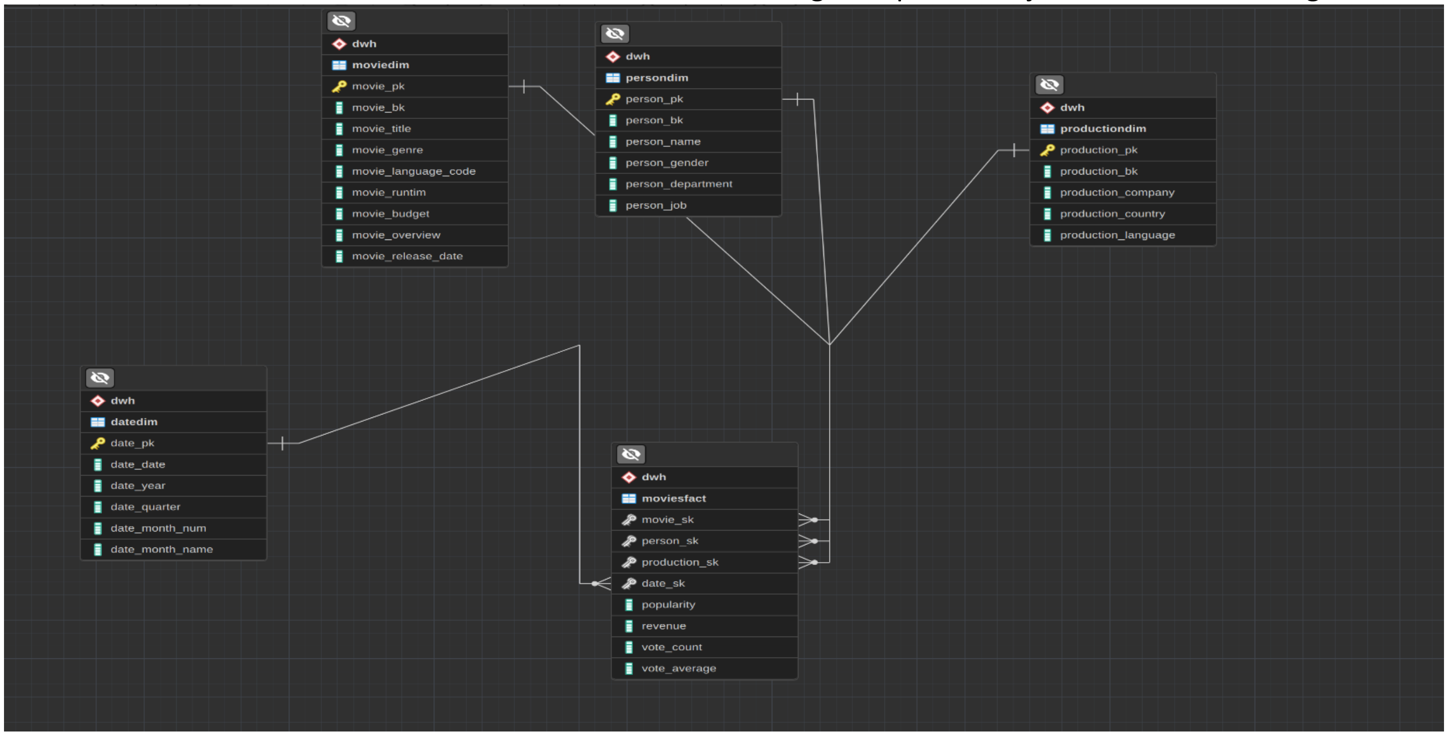
After Running Dag



MoviesStagingArea
Collections (15)
country
department
gender
genre
language
language_role
movie
movie_cast
movie_company
movie_crew
movie_genres
movie_languages
person
production_company
production_country

VII- Movies Production Insights Pipeline Project Data Warehouse

The data warehouse for the Movies Production Insights Pipeline Project is structured using a



denormalized star schema, optimizing the schema for efficient querying and analysis of movie-related data. The schema includes four dimension tables and one fact table, each playing a crucial role in providing comprehensive insights into movie production data.

1. Movie Dimension (MovieDim):

- Surrogate Key (movie_sk):
  - Unique identifier for each movie record in the dimension table.
- Business Key (movie_bk):
  - Business identifier for each movie, used for joining with other tables.
- Movie Title (movie_title):
  - The title of the movie.
- Movie Genre (movie_genre):
  - The genre of the movie.
- Movie Language Code (movie_language_code):
  - Language code of the movie.
- Movie Runtime (movie_runtime):
  - Duration of the movie in minutes.
- Movie Budget (movie_budget):
  - Budget allocated for the movie.
- Movie Overview (movie_overview):
  - Brief overview or summary of the movie.
- Movie Release Date (movie_release_date):
  - The date when the movie was released.

2. Person Dimension (PersonDim):

- Surrogate Key (person_sk):
    - Unique identifier for each person record in the dimension table.
- Business Key (person_bk):
    - Business identifier for each person, used for joining with other tables.
- Person Name (person_name):
    - Full name of the person.
- Person Gender (person_gender):
    - Gender of the person.
- Person Department (person_department):
    - Department or role of the person.
- Person Job (person_job):
    - Job or position of the person.

3. Production Dimension (ProductionDim):

- Surrogate Key (production_sk):
    - Unique identifier for each production record in the dimension table.
- Business Key (production_bk):
    - Business identifier for each production, used for joining with other tables.
- Production Company (production_company):
    - Company responsible for the movie production.
- Production Country (production_country):
    - Country where the movie was produced.
- Production Language (production_language):
    - Language used in the production.

4. Date Dimension (DimDate):

- Surrogate Key (date_sk):
    - Unique identifier for each date record in the dimension table.
- Date (date_date):
    - Calendar date.
- Year (date_year):
    - The year of the date.
- Quarter (date_quarter):
    - The quarter of the year (e.g., Q1, Q2, Q3, Q4).
- Month Number (date_month_num):
    - Numeric representation of the month.
- Month Name (date_month_name):
    - The name of the month.
- Day (date_day):
    - The day of the month.

5. Movies Fact (MoviesFact):

- Surrogate Key (movie_sk):
    - Foreign key referencing the MovieDim.
- Person Surrogate Key (person_sk):

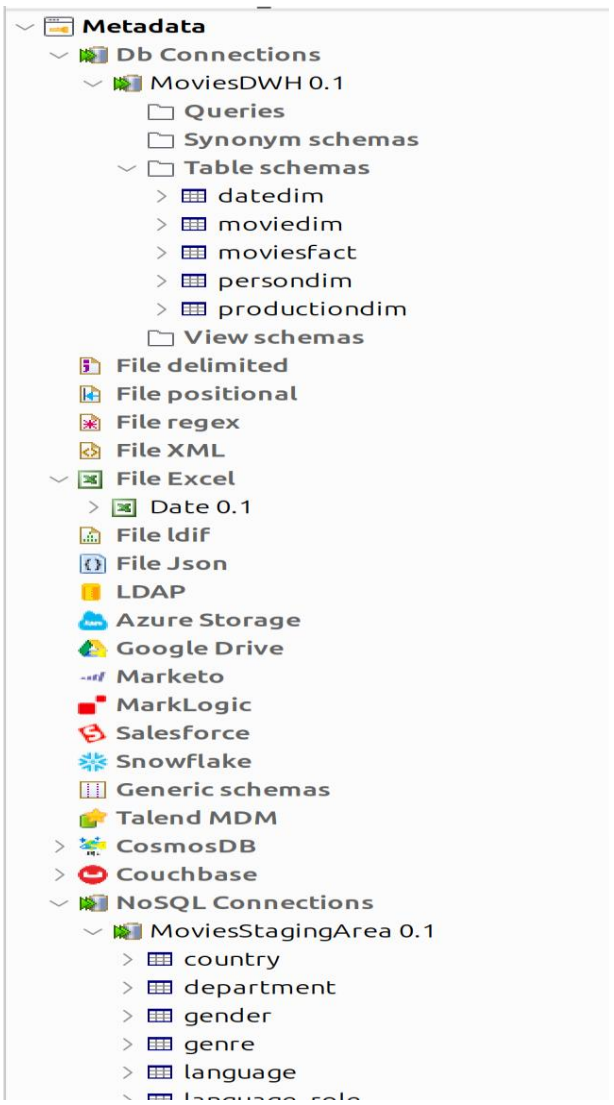
    - Foreign key referencing the PersonDim.
- Production Surrogate Key (production_sk):
    - Foreign key referencing the ProductionDim.
- Date Surrogate Key (date_sk):
    - Foreign key referencing the DimDate.
- Popularity (popularity):
    - Popularity score of the movie.
- Revenue (revenue):
    - Revenue generated by the movie.
- Vote Count (vote_count):
    - Count of votes received by the movie.
- Vote Average (vote_average):
    - Average vote rating given to the movie.

Schema Relationships:

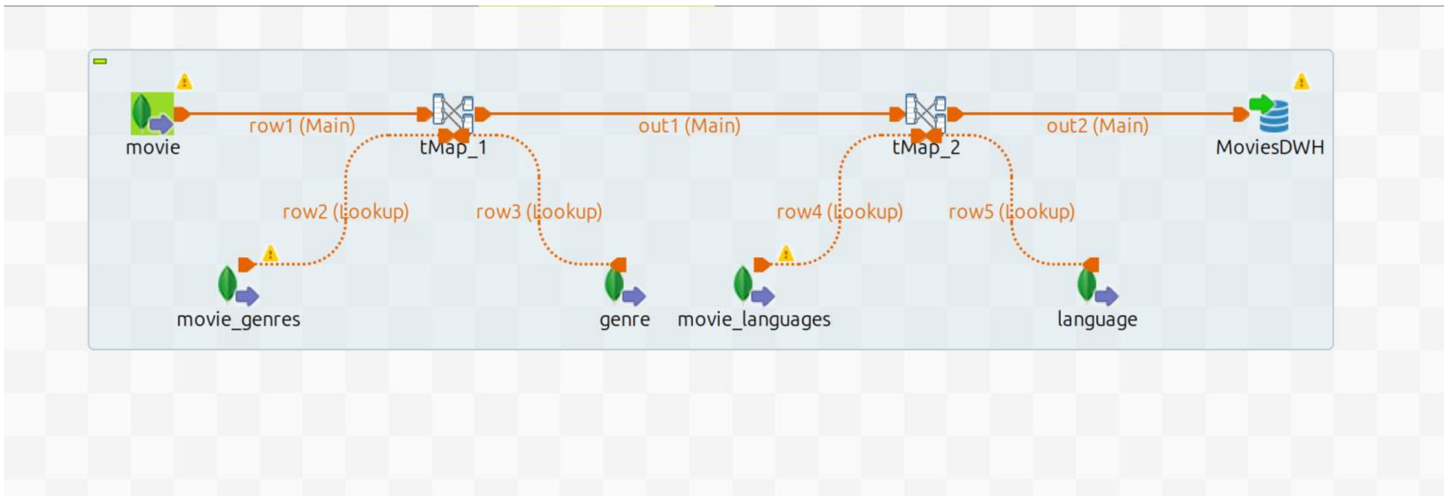
- The fact table (MoviesFact) connects to each dimension table through surrogate keys, allowing for efficient querying and analysis.
- Dimension tables provide detailed attributes related to movies, persons, productions, and dates.
- The star schema facilitates simplified and optimized queries for insights into movie production data.

VIII- Data Transformation using Talend (ETL) to Data Warehouse

1. Talend Meta Data:


Metadata
Db Connections
MoviesDWH 0.1
Queries
Synonym schemas
Table schemas
datedim
moviedim
moviesfact
persondim
productiondim
View schemas
File delimited
File positional
File regex
File XML
File Excel
Date 0.1
File ldif
File Json
LDAP
Azure Storage
Google Drive
Marketo
MarkLogic
Salesforce
Snowflake
Generic schemas
Talend MDM
CosmosDB
Couchbase
NoSQL Connections
MoviesStagingArea 0.1
country
department
gender
genre
language

2. Load Movie Dimension (MovieDim):



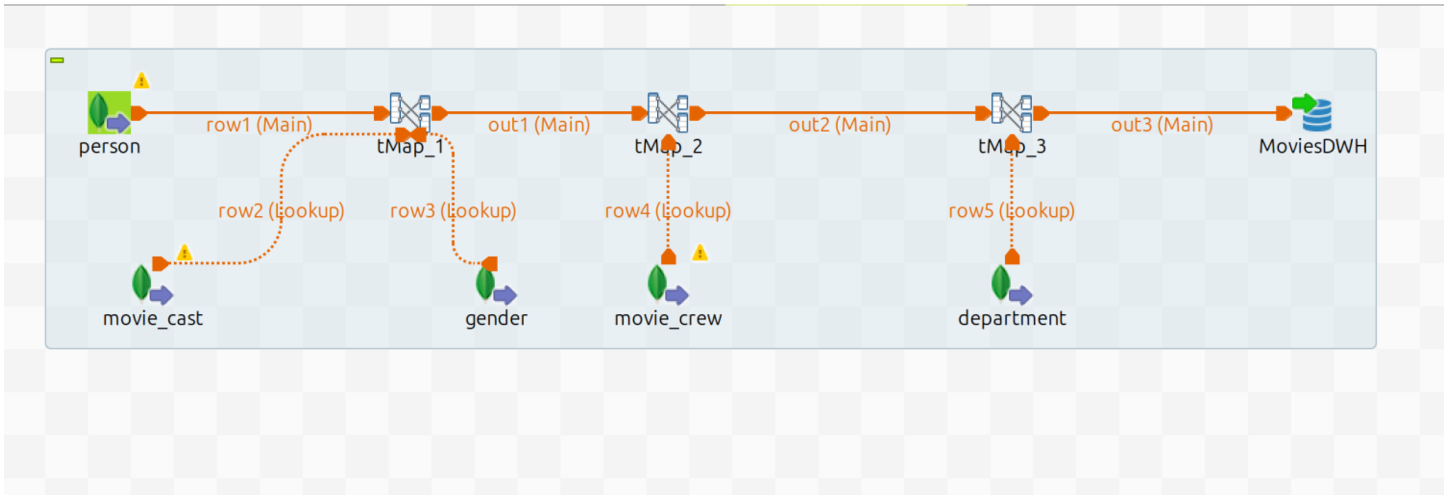
To load the Movie Dimension, a series of joins and transformations were performed using the following tables:

- Movie:
  - Primary source for movie details.
- Movie Genres:
  - Contains genre information for movies.
- Genre:
  - Provides genre details.
- Movie Language:
  - Includes language information for movies.
- Language:
  - Contains language details.

Transformation Steps:

- Join Movie and Movie Genres:
  - Utilize the movie and movie genres tables to associate movies with their respective genres.
- Join with Genre Table:
  - Incorporate genre details into the dataset.
- Join with Movie Language:
  - Include language information for each movie.
- Join with Language Table:
  - Enhance the dataset with detailed language attributes.
- Load to Data Warehouse:
  - Utilize Talend to load the transformed data into the Movie Dimension table.

3. Load Person Dimension (PersonDim):



Loading the Person Dimension involves integrating data from various tables, including:

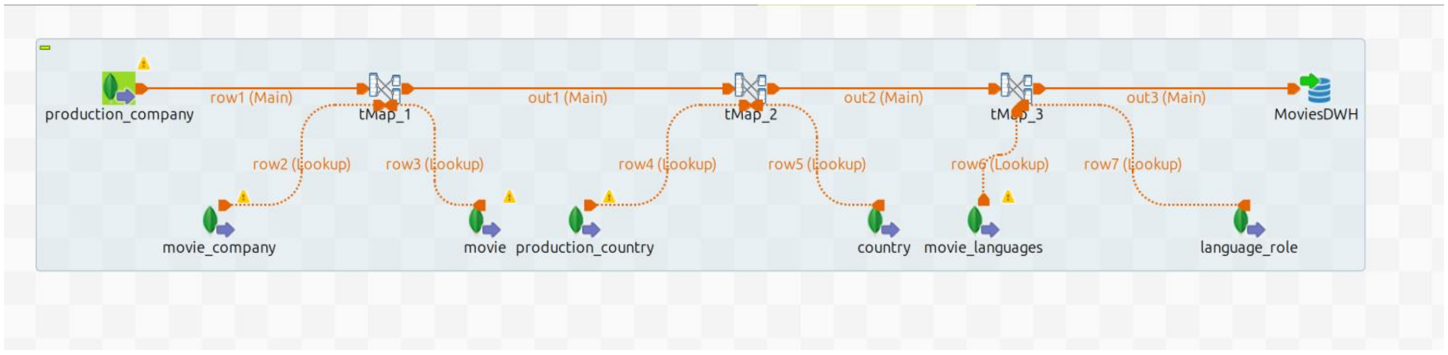
- Person:
  - Primary source for person details.
- Movie Cast:
  - Contains information about movie casts.

- Gender:
  - Provides gender details.
- Movie Crew:
  - Includes information about movie crew members.
- Department:
  - Contains details about different departments.

Transformation Steps:

- Join Person and Movie Cast:
  - Associate persons with movie cast details.
- Join with Gender Table:
  - Incorporate gender details for each person.
- Join with Movie Crew:
  - Include information about movie crew members.
- Join with Department Table:
  - Enhance the dataset with department attributes.
- Load to Data Warehouse:
  - Utilize Talend to load the transformed data into the Person Dimension table.

4. Load Production Dimension (ProductionDim):



To load the Production Dimension, data is extracted and transformed from the following tables:

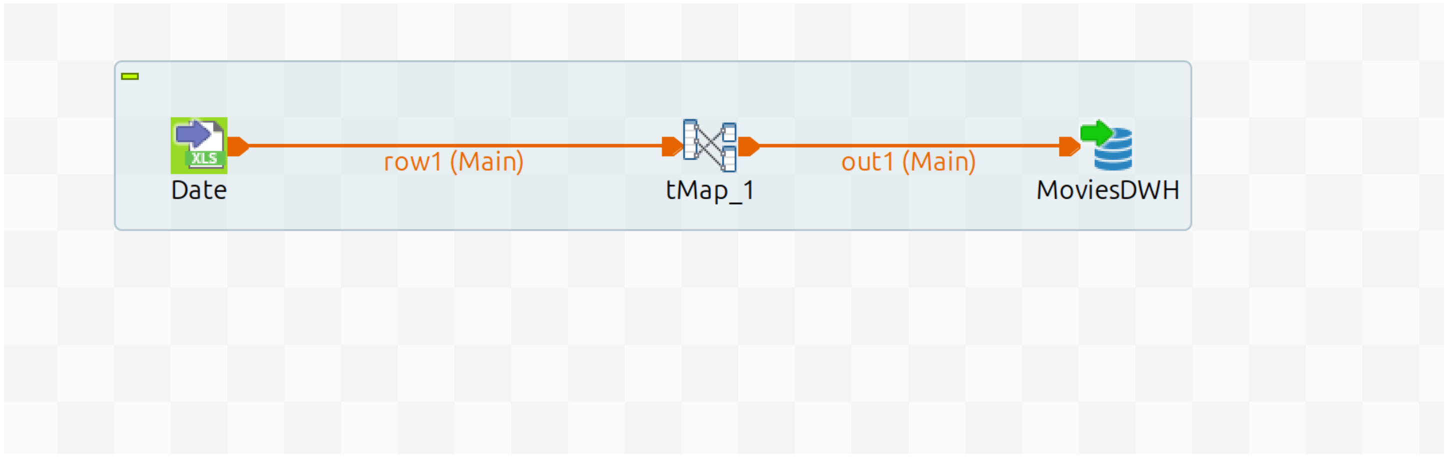
- Production Company:
  - Primary source for production company details.
- Movie Company:
  - Contains information about movie companies.
- Movie:
  - Primary source for movie details.
- Production Country:
  - Includes information about production countries.
- Country:
  - Contains country details.
- Movie Language:
  - Includes language information for movies.
- Language Role:

- Contains language role details.

Transformation Steps:

- Join Production Company and Movie Company:
  - Associate production companies with movie companies.
- Join with Movie Table:
  - Incorporate movie details into the dataset.
- Join with Production Country:
  - Include information about production countries.
- Join with Country Table:
  - Enhance the dataset with detailed country attributes.
- Join with Movie Language and Language Role:
  - Incorporate language details and language roles for each movie.
- Load to Data Warehouse:
  - Utilize Talend to load the transformed data into the Production Dimension table.

5. Load Date Dimension (DimDate):

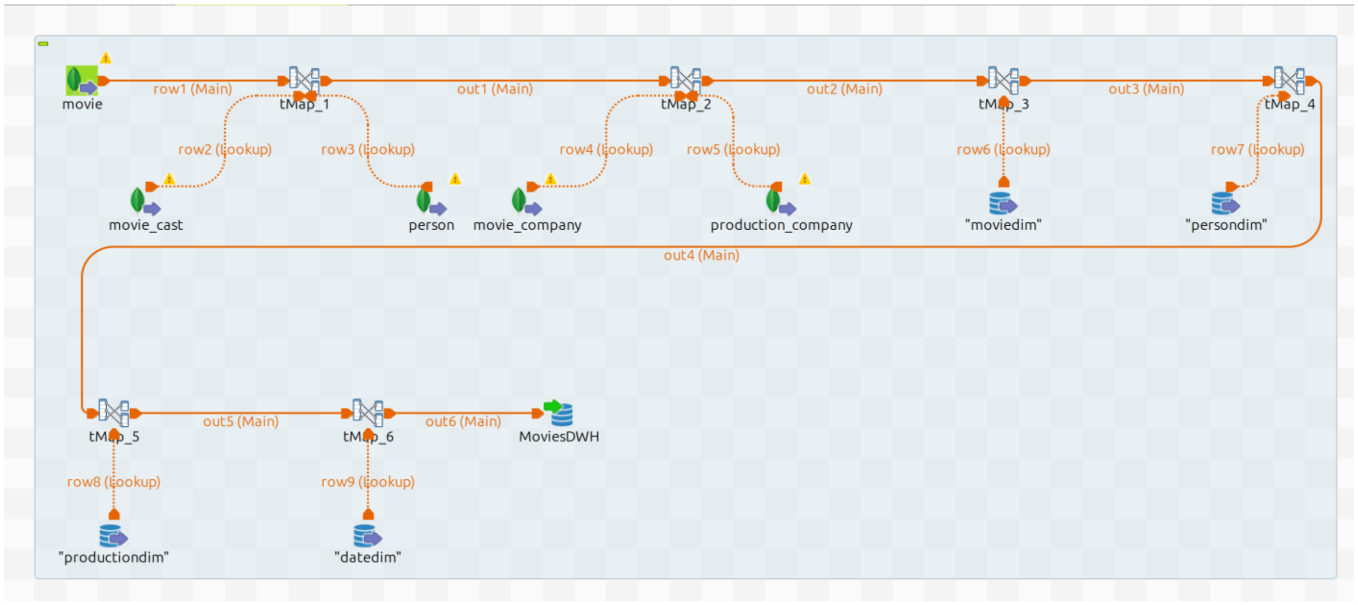


The Date Dimension is loaded using an XLS file containing date information.

Transformation Steps:

- Read XLS File:
  - Read the XLS file containing date information.
- Load to Data Warehouse:
  - Utilize Talend to load the date data into the Date Dimension table.

6. Load Fact Table (MoviesFact):



Loading the Fact Table involves complex joins and lookups:

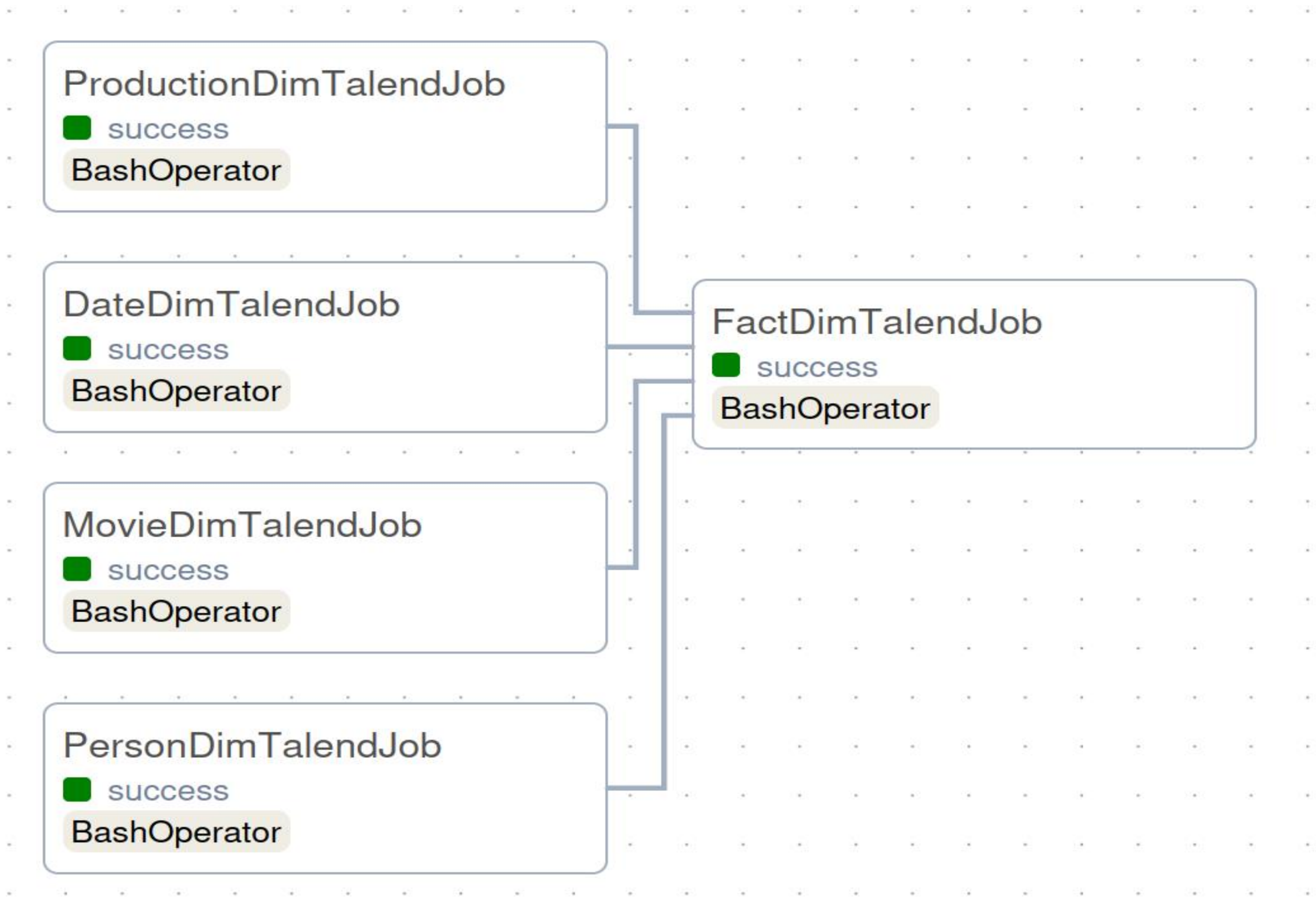
- Movie:
  - Primary source for movie details.

- Movie Cast:
  - Contains information about movie casts.
- Person:
  - Primary source for person details.
- Movie Company:
  - Contains information about movie companies.
- Production Company:
  - Primary source for production company details.
- Lookups (MovieDim, PersonDim, ProductionDim, DateDim):
  - Tables containing surrogate keys for dimensions.

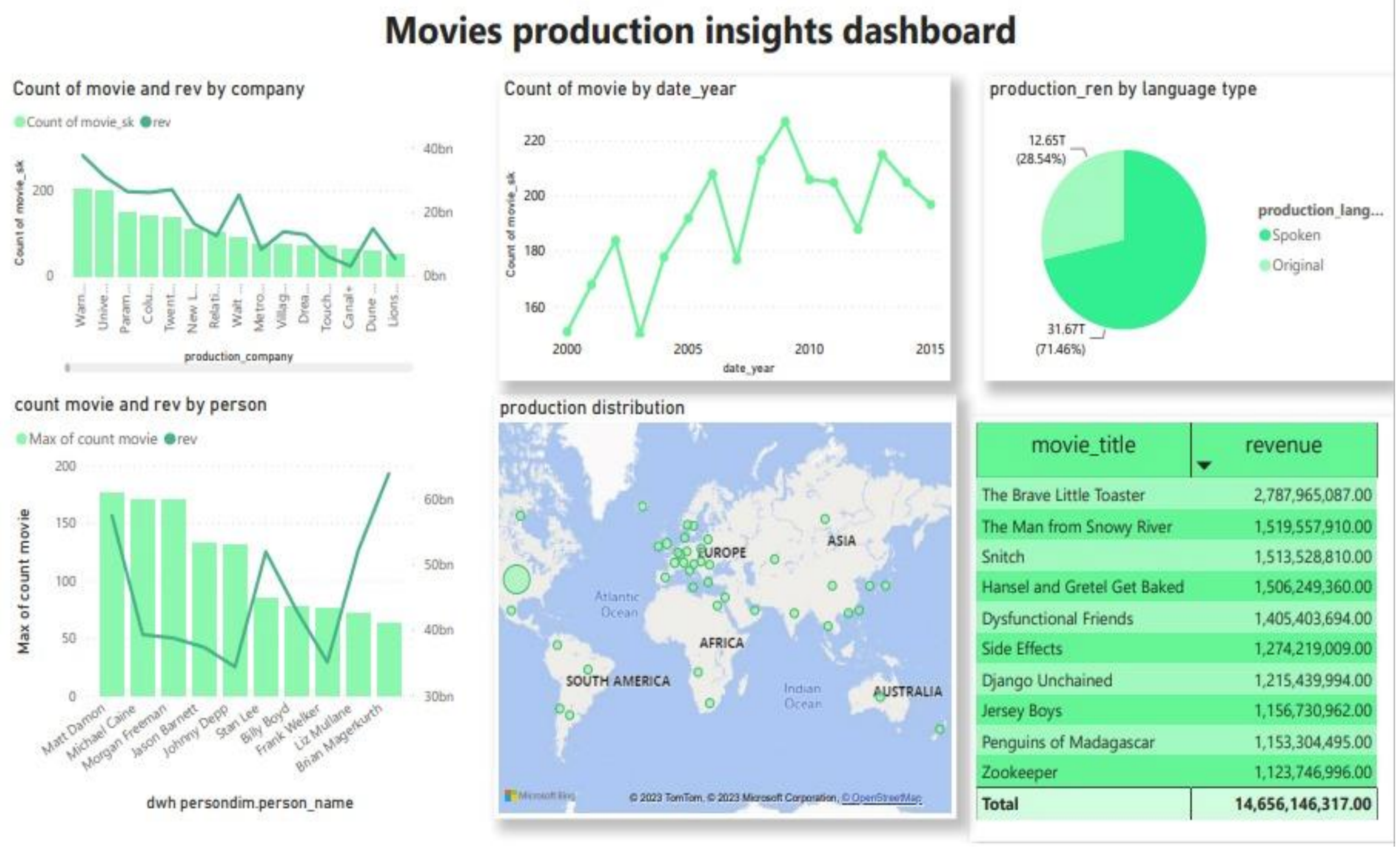
Transformation Steps:

- Join Movie, Movie Cast, Person, Movie Company, Production Company:
  - Associate movies with cast, persons, movie companies, and production companies.
- Perform Lookups for Surrogate Keys:
  - Utilize lookup tables for MovieDim, PersonDim, ProductionDim, and DateDim to obtain surrogate keys.
- Load to Data Warehouse:
  - Utilize Talend to load the transformed data into the Fact Table.

Second Dag

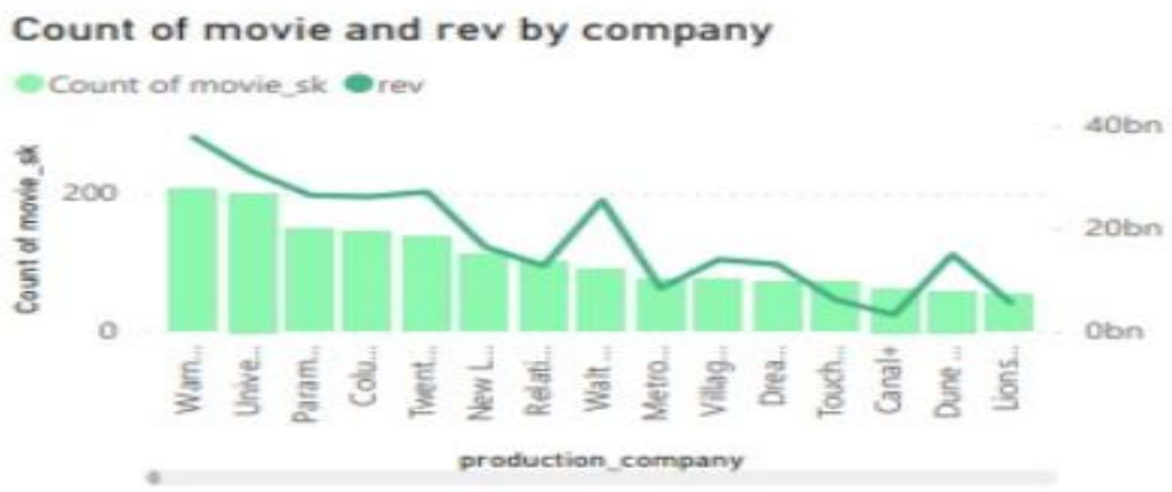


IX- Movies Insights Dashboard



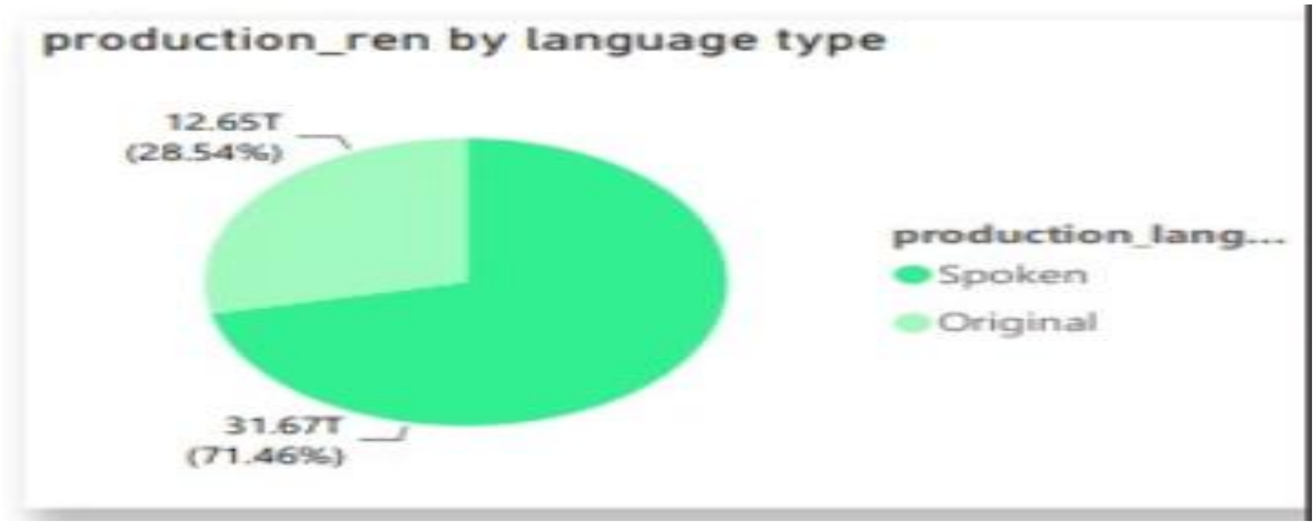
movie_title	revenue
The Brave Little Toaster	2,787,965,087.00
The Man from Snowy River	1,519,557,910.00
Snitch	1,513,528,810.00
Hansel and Gretel Get Baked	1,506,249,360.00
Dysfunctional Friends	1,405,403,694.00
Side Effects	1,274,219,009.00
Django Unchained	1,215,439,994.00
Jersey Boys	1,156,730,962.00
Penguins of Madagascar	1,153,304,495.00
Zookeeper	1,123,746,996.00
Total	**14,656,146,317.00**

- **Count Of Movie and Rev by Company :**



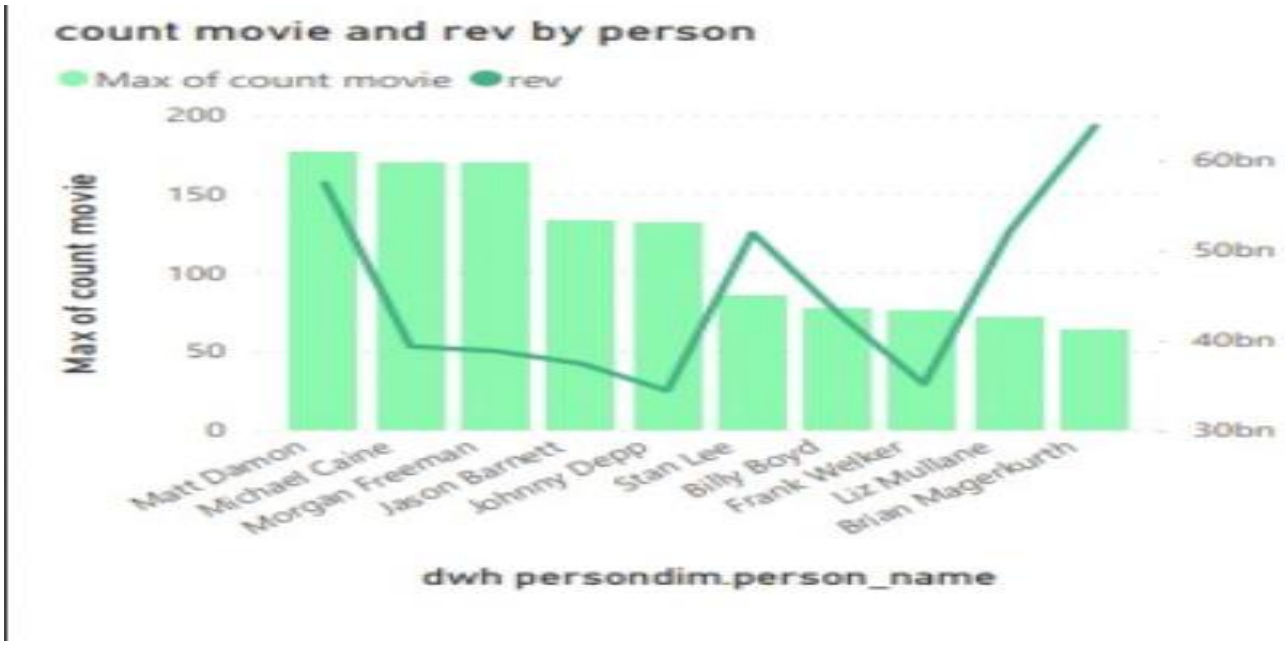
  - Analysis of the relationship between companies' profits and the number of films produced

- **Production revenue by Language Type :**



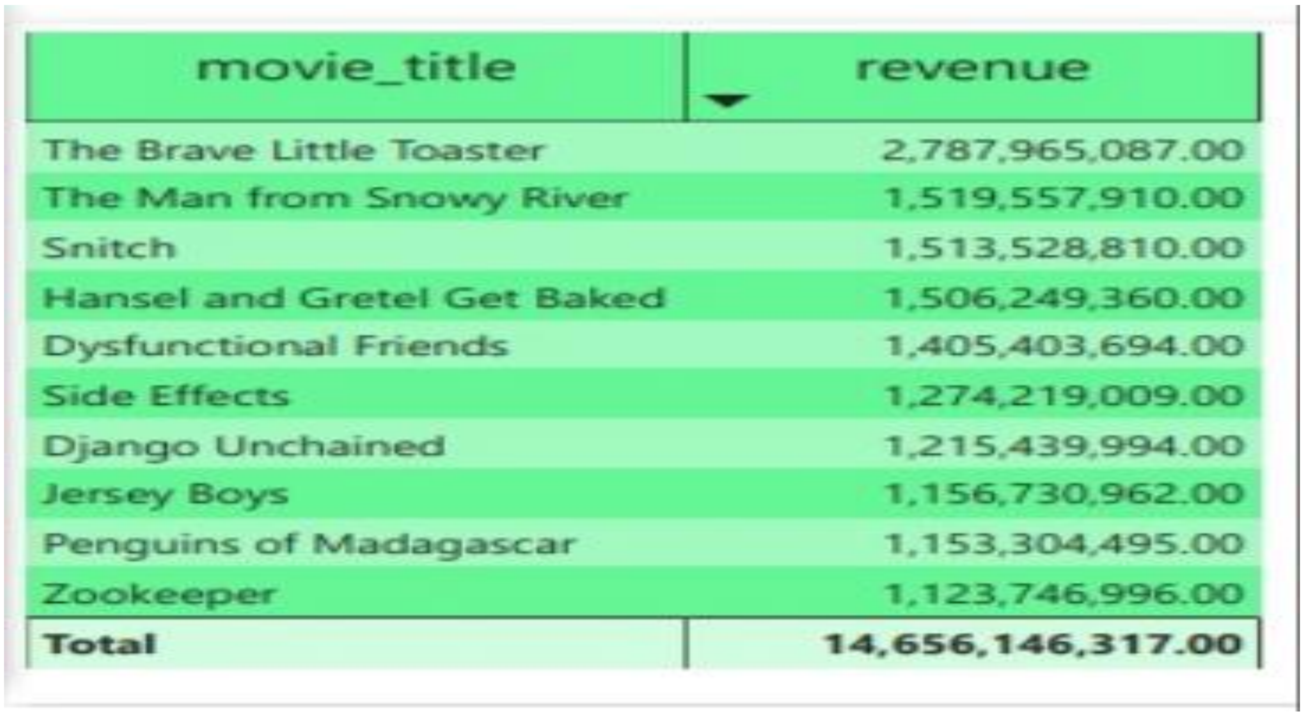
  - The profit of companies that use dubbing into other languages compared to the profit of companies that only use their original language

- **Count Movie and Revenue By Person :**



- The impact of the actor/director on the profits of films compared to relying on all the films mentioned in them

- **Movie and Revenue :**



movie_title	revenue
The Brave Little Toaster	2,787,965,087.00
The Man from Snowy River	1,519,557,910.00
Snitch	1,513,528,810.00
Hansel and Gretel Get Baked	1,506,249,360.00
Dysfunctional Friends	1,405,403,694.00
Side Effects	1,274,219,009.00
Django Unchained	1,215,439,994.00
Jersey Boys	1,156,730,962.00
Penguins of Madagascar	1,153,304,495.00
Zookeeper	1,123,746,996.00
Total	**14,656,146,317.00**

- Describe the Movie name and it’s Revenue

X- <u>Conclusion</u>

<u>Project Objectives:</u>

- The project aimed to address critical aspects of the movie industry through:

Data Integration:

- Consolidating diverse data sources related to persons, movies, genres, production, etc., into a centralized data warehouse.

Automation:

- Implementing Apache Airflow for automated orchestration of data pipelines to ensure timely and accurate data processing.

Transformation:

- Leveraging Spark for effective data extraction and transformation, converting raw data into a structured format suitable for analysis.

Staging Area:

- Implementing MongoDB as a NoSQL staging area for interim data storage, providing flexibility and scalability.

ETL Process:

- Utilizing Talend for Extract, Transform, Load (ETL) processes to seamlessly transfer data from the staging area to the PostgreSQL data warehouse.

Change Data Capture (CDC):

- Implementing a Change Data Capture (CDC) strategy using the Meta Data table to capture and track changes, ensuring the data warehouse reflects the latest information.

Visualization:

- Integrating Power BI for insightful visualization, enabling users to explore trends, patterns, and relationships within the movie industry.

Ecosystem Solution:

- The project employed a comprehensive ecosystem solution, incorporating technologies such as Apache Airflow, Spark, MongoDB, PostgreSQL, Talend, and Power BI. This combination of tools facilitated a streamlined and efficient data pipeline, covering data extraction, transformation, loading, and visualization.

Data Sources:

- Data from diverse sources, including relational databases and external files in formats such as XML, JSON, and XLS, were integrated to create a coherent dataset. Key tables, including Person, Movie Cast, Movie, Production Company, and others, formed the core of the dataset.

Data Warehouse Structure:

- The data warehouse adopted a denormalized star schema, optimizing the schema for efficient querying and analysis of movie-related data. It comprised four dimension tables (MovieDim, PersonDim, ProductionDim, DimDate) and one fact table (MoviesFact). The relationships among these tables facilitated simplified and optimized queries, providing comprehensive insights into movie production data.

Visualization of Insights:

- Insights derived from the data warehouse were visualized using Power BI. The Movies Insights Dashboard presented key metrics, including the count of movies and revenue by company, production revenue by language type, count of movies and revenue by person, and individual movie details with revenue.

Conclusion:

- In conclusion, the Movies Production Insights Pipeline Project has successfully established a robust data infrastructure for the movie industry, allowing stakeholders to make informed decisions based on comprehensive and up-to-date information. The combination of diverse technologies, effective data transformation processes, and insightful visualizations positions this project as a valuable asset in navigating the complexities of the dynamic movie industry. The implemented data pipeline not only centralizes information but also ensures the scalability and adaptability needed to meet evolving industry demands.