Microsoft Fabric - OTT Data Engineering Project (Movie Datasets)

Project Overview

Project Description and Purpose

This project aims to develop a comprehensive **ETL pipeline** for an **OTT platform** (**Movie Datasets**) using **Microsoft Fabric**. The objective is to extract, transform, and load (ETL) movie-related data into a structured format to facilitate **data analysis and reporting**.

Purpose:

- Enable **automated data ingestion** from on-premise sources using **Power BI Gateway**.
- Transform raw data into structured data using **PySpark Notebooks**
- Store and manage structured data in **Microsoft Fabric Lakehouse & Warehouse**.
- Implement a **star schema** for optimized querying and reporting.
- Develop **Power BI dashboards** to provide insights into **movie trends, user engagement, and genre performance**.

Source Data Description

The dataset used in this project was downloaded from Kaggle: TMDB Movies Dataset (2023).

Dataset Fields:

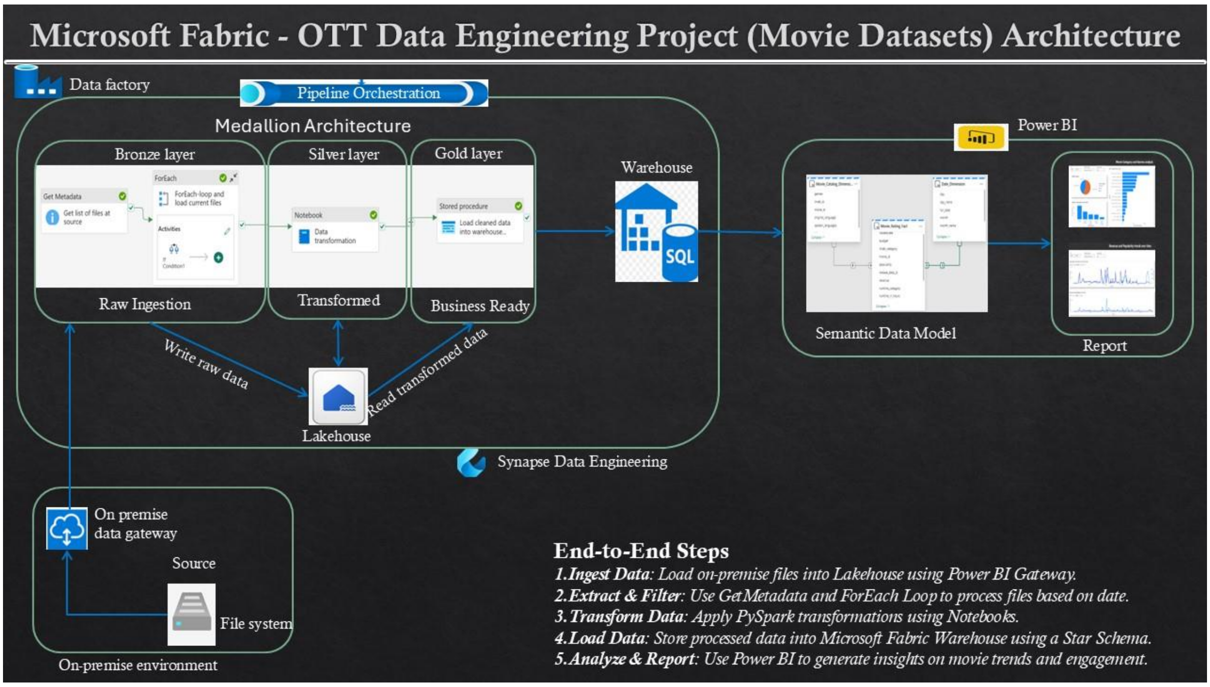
- **id**: Unique identifier for each movie. *(int)*
- **title**: Title of the movie. *(str)*
- **vote_average**: Average vote or rating given by viewers. *(float)*
- **vote_count**: Total count of votes received for the movie. *(int)*
- **status**: The status of the movie (e.g., Released, Rumored, Post Production, etc.). *(str)*
- **release_date**: Date when the movie was released. *(str)*
- **revenue**: Total revenue generated by the movie. *(int)*
- **runtime**: Duration of the movie in minutes. *(int)*
- **adult**: Indicates if the movie is suitable only for adult audiences. *(bool)*
- **budget**: Budget allocated for the movie. *(int)*
- **imdb_id**: IMDb ID of the movie. *(str)*
- **original_language**: Original language in which the movie was produced. *(str)*
- **original_title**: Original title of the movie. *(str)*
- **popularity**: Popularity score of the movie. *(float)*
- **genres**: List of genres the movie belongs to. *(str)*
- **spoken_languages**: List of languages spoken in the movie. *(str)*

This project involves building an end-to-end ETL pipeline for an OTT platform (Netflix, Prime, Disney+, etc.) using **Microsoft Fabric**. The goal is to ingest on-premise data, transform it, and load it into a structured format for reporting and analysis in **Power BI**.

Services Used in Microsoft Fabric

- **OneLake** (Unified storage)
- **Data Factory** (Data Ingestion)
- **Lakehouse** (Storage & Querying)
- **Warehouse** (Data Modeling & Analytics)
- **Notebooks** (Transformation using Spark/PySpark)
- **Power BI** (Visualization & Reporting)

Project Architecture



End-to-End Steps

*1.**Ingest Data**: Load on-premise files into Lakehouse using Power BI Gateway.*
*2.**Extract & Filter**: Use GetMetadata and ForEach Loop to process files based on date.*
*3.**Transform Data**: Apply PySpark transformations using Notebooks.*
*4.**Load Data**: Store processed data into Microsoft Fabric Warehouse using a Star Schema.*
*5.**Analyze & Report**: Use Power BI to generate insights on movie trends and engagement.*

End-to-End Steps

1. **Ingest Data**: Load on-premise files into OneLake using Power BI Gateway.
2. **Extract & Filter**: Use GetMetadata and ForEach Loop to process files based on date.
3. **Transform Data**: Apply PySpark transformations using Notebooks.
4. **Load Data**: Store processed data into Microsoft Fabric Warehouse using a Star Schema.
5. **Analyze & Report**: Use Power BI to generate insights on movie trends and engagement.

Data Flow

1. **Ingestion**: Load on-premise file storage data into **OneLake** using **Power BI Gateway**.
2. **Transformation**: Process and clean data using **Notebooks** and **Stored Procedures**.
3. **Storage**: Store refined data in **Lakehouse tables (Silver Layer)**.
4. **Warehouse & Modeling**: Load structured data into **Fabric Warehouse** and create a **Star Schema**.

5. **Reporting**: Use **Power BI** to visualize and analyze trends in OTT platform performance.

Step-by-Step Implementation

1. Environment Setup

1. Create a **Microsoft Fabric Workspace**.
2. Set up **OneLake Storage** for raw and refined data.
3. Configure **Power BI Gateway** for accessing on-premise data.
4. Create a **Lakehouse** to store structured data.
5. Create a **Warehouse** to manage structured data modeling.
6. Enable **Power BI integration** for reporting.

2. Data Ingestion

- **Source**: On-premise files (CSV, JSON, Parquet)
- **Destination**: Microsoft Fabric **OneLake (Bronze Layer)**
- **Pipeline**:
  - **Setup Power BI Gateway** to connect to on-prem data.
  - Use **Copy Activity** to fetch data based on filename.
  - Extract filenames using **GetMetadata Activity**.
  - Pass filenames into a **ForEach loop**.
  - Use an **If Condition** to check if the filename matches the current date.
  - Call **Copy Activity** to load the matching file into **Lakehouse (Silver Layer)**.

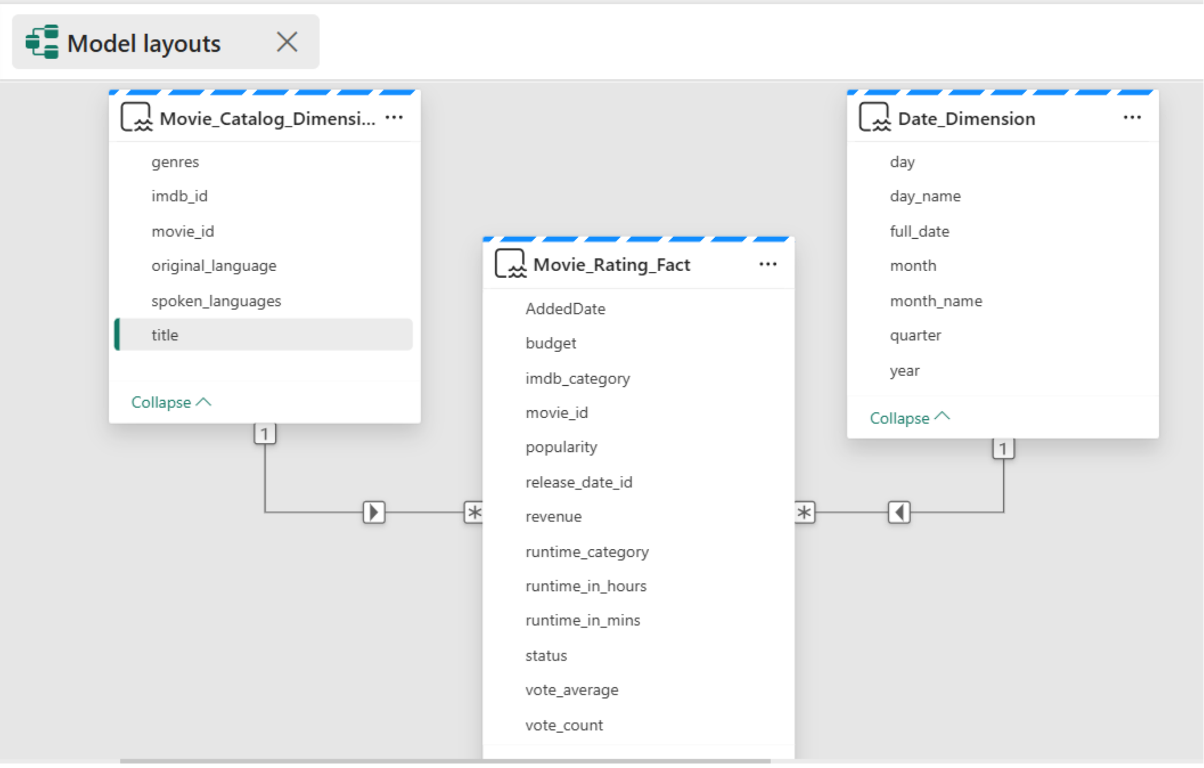
3. Data Transformation (ETL)

- Use **Notebooks in Fabric** to transform data with **PySpark**.
- Transformations include:
  - **Handling NULL values**
  - **Removing duplicates**
  - **Deriving new columns** (e.g., IMDB rating categories, runtime classification)
  - **Standardizing date formats**
- Save the processed data in **Lakehouse (Silver Layer)**.

4. Data Loading to Warehouse

- Load transformed data from **Lakehouse (Silver Layer)** into **Microsoft Fabric Warehouse**.
- Create a **Star Schema** in the Warehouse:
- Use **Stored Procedure Activity** to execute transformation logic.

Star Schema



SQL Schema Design for Microsoft Fabric Warehouse - Movies

1. Date Dimension Table

```
CREATE TABLE Movies.Date_Dimension (
    full_date DATE ,
    year INT NOT NULL,
    month INT NOT NULL,
    day INT NOT NULL,
    day_name VARCHAR(20),
    month_name VARCHAR(20),
    quarter INT
);
```

2. Movie Catalog Dimension Table

```
CREATE TABLE Movies.Movie_Catalog_Dimension (
    movie_id INT,
    title VARCHAR(255) ,
    imdb_id VARCHAR(20),
    original_language VARCHAR(10),
    genres VARCHAR(255),
    spoken_languages VARCHAR(255)
);
```

3. **Movie Rating Fact Table**

```
CREATE TABLE Movies.Movie_Rating_Fact (
    movie_id INT NOT NULL,
    vote_average FLOAT NOT NULL,
    vote_count INT NOT NULL,
    status VARCHAR(250),
    release_date_id DATE ,
    runtime_in_mins INT,
    runtime_in_hours FLOAT,
    runtime_category VARCHAR(250),
    budget BIGINT,
    revenue BIGINT,
    popularity FLOAT,
    imdb_category VARCHAR(250),
    AddedDate DATE
);
```

5. Stored Procedure for Daily Updates

- **Stored Procedure Activity** is executed in a Microsoft Fabric pipeline.
- **MERGE** is used for **Dimension Tables**.
- **APPEND** is used for the **Fact Table**.

6. Execution in Pipeline

- Ensures new records are inserted while avoiding duplicates.
- Schema Relationships:
  - **Fact Table** references **Movie_Catalog_Dimension** via `movie_id` and **Date_Dimension** via `release_date_id`.

7. Reporting in Power BI

- Connect **Power BI** to the Fabric **Warehouse**.
- Create dashboards to analyze:

1. Top-performing genres:

- Use the **Movie Rating Fact Table** and **Movie Catalog Dimension**.
- Aggregate **vote_count** and **popularity** to determine the highest-rated genres.
- Create a **bar chart** to display genre performance.

2. Trends in movie viewership:

- Utilize the **Date Dimension** and **Movie Rating Fact Table**.
- Track **popularity** and **view counts** over time using a **line chart**.
- Apply filters for **yearly, monthly, and weekly trends**.

3. User engagement insights:

- Combine **Movie Rating Fact Table** with **Date Dimension**.
- Analyze **runtime categories** and their impact on engagement.

- Publish reports to the **Fabric Workspace** and set up scheduled refresh for real-time insights.
- Connect **Power BI** to the Fabric **Warehouse**.
- Create dashboards to analyze:
    - Top-performing genres
    - Trends in movie viewership
    - User engagement insights
- Publish reports to the **Fabric Workspace**.

Conclusion

This project successfully migrated an **ADF + Synapse** ETL workflow to **Microsoft Fabric**, incorporating **Warehouse capabilities** for structured data modelling and efficient querying.
