Homework Lab

Ilhan Ankoue & Lucas Mastier

Analyse des comédies musicales les mieux notées et leur évolution au fil du temps

1. Objectif du Projet

Le projet vise à analyser les comédies musicales les mieux notées et à suivre l'évolution de leurs notes au fil du temps. L'analyse s'appuie sur des données issues de critiques d'utilisateurs. Nous avons utilisé Azure pour réaliser ce projet.

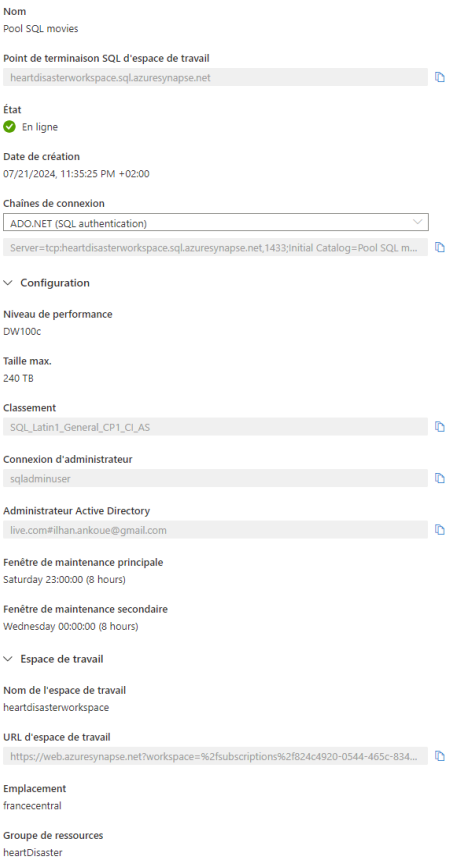
2. Nettoyage des Données

Pour garantir l'intégrité des données, nous avons nettoyé le fichier CSV user_reviews en supprimant toutes les lignes où la colonne reviewId était NULL. Nous avons utilisé un script Python pour accomplir cette tâche :

```
import pandas as pd

# Lire le fichier CSV
df = pd.read_csv(r"C:\Users\U1\Downloads\archive\user_reviews.csv\user_reviews.csv")

# Supprimer les lignes où reviewId est NULL ou vide
df_cleaned = df[df['reviewId'].notna() & (df['reviewId'] != '')]

# Sauvegarder le fichier nettoyé
df_cleaned.to_csv(r"C:\Users\U1\Downloads\archive\user_reviews.csv\user_reviews_clean.csv", index=False)
```

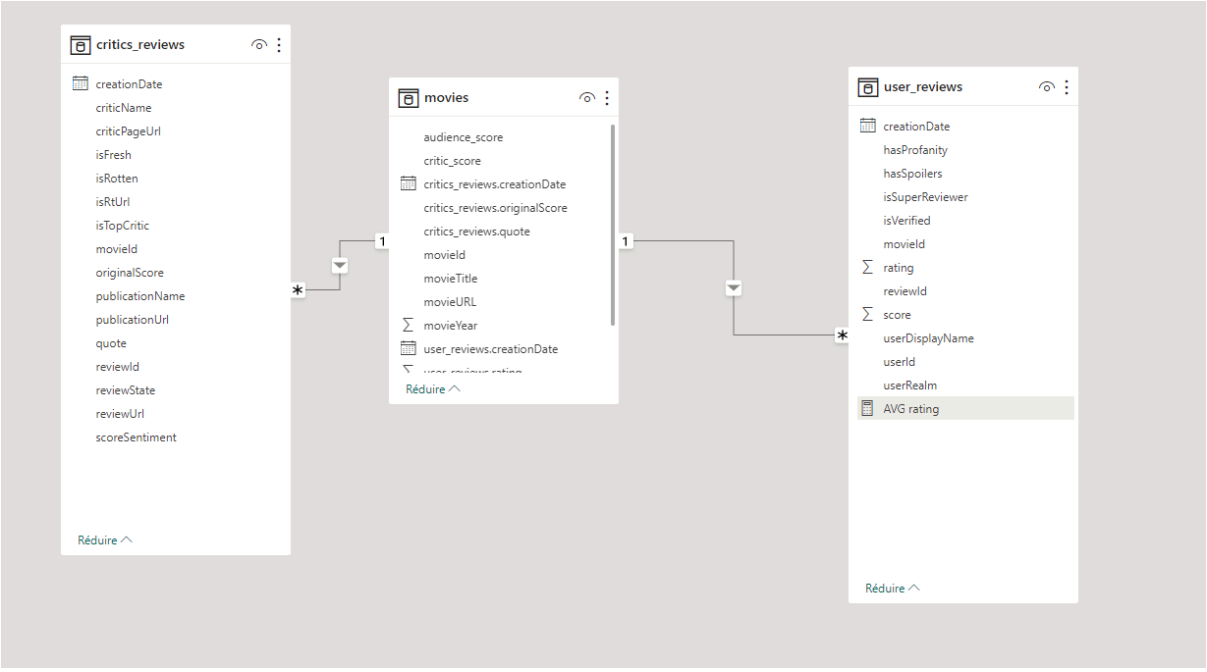
3. Création de l'Infrastructure sur Azure

- **Azure Data Lake Storage Gen2** : Nous avons créé un Data Lake pour stocker nos fichiers CSV provenant d'un dataset Kaggle.
- **Groupe de Ressources et Compte de Stockage** : Mise en place d'un groupe de ressources et d'un compte de stockage sur Azure pour gérer nos actifs.
- **Azure Synapse Analytics** : Création d'un environnement Synapse et d'un SQL pool pour le traitement des données.



4. Création et Remplissage des Tables dans Synapse

1. Création des Tables



1.1 Entités et Attributs

1. critics_reviews
    - reviewId (PK) : Identifiant unique de la critique
    - movieId : Identifiant du film (FK)
    - creationDate : Date de création de la critique
    - criticName : Nom du critique
    - criticPageUrl : URL de la page du critique
    - reviewState : État de la critique
    - isFresh : Indicateur si la critique est positive
    - isRotten : Indicateur si la critique est négative
    - isRtUrl : URL vers Rotten Tomatoes
    - isTopCritic : Indicateur si le critique est un critique top
    - publicationUrl : URL de la publication
    - publicationName : Nom de la publication
    - reviewUrl : URL de la critique
    - quote : Citation de la critique
    - scoreSentiment : Sentiment du score
    - originalScore : Score original de la critique
2. movies
    - movieId (PK) : Identifiant unique du film
    - movieTitle : Titre du film
    - movieYear : Année de sortie du film
    - movieURL : URL du film
    - critic_score : Score des critiques
    - audience_score : Score du public
3. user_reviews
    - reviewId (PK) : Identifiant unique de la critique
    - movieId : Identifiant du film (FK)
    - rating : Note attribuée par l'utilisateur
    - quote : Citation de l'utilisateur
    - isVerified : Indicateur si l'utilisateur est vérifié
    - isSuperReviewer : Indicateur si l'utilisateur est un super critique
    - hasSpoilers : Indicateur si la critique contient des spoilers
    - hasProfanity : Indicateur si la critique contient des obscénités
    - score : Score attribué par l'utilisateur
    - creationDate : Date de création de la critique
    - userDisplayName : Nom d'affichage de l'utilisateur
    - userRealm : Domaine de l'utilisateur
    - userId : Identifiant unique de l'utilisateur

Relations

1. critics_reviews a une relation N-1 avec movies:
    - Chaque critique de critics_reviews est liée à un film dans movies.
    - movieId dans critics_reviews est une clé étrangère (FK) référencée à movieId dans movies.
2. user_reviews a une relation N-1 avec movies:
    - Chaque critique d'utilisateur de user_reviews est liée à un film dans movies.

- movieId dans user_reviews est une clé étrangère (FK) référencée à movieId dans movies.

1.2 Création des tables

Nous avons créé trois tables dans Azure Synapse Analytics pour stocker nos données :

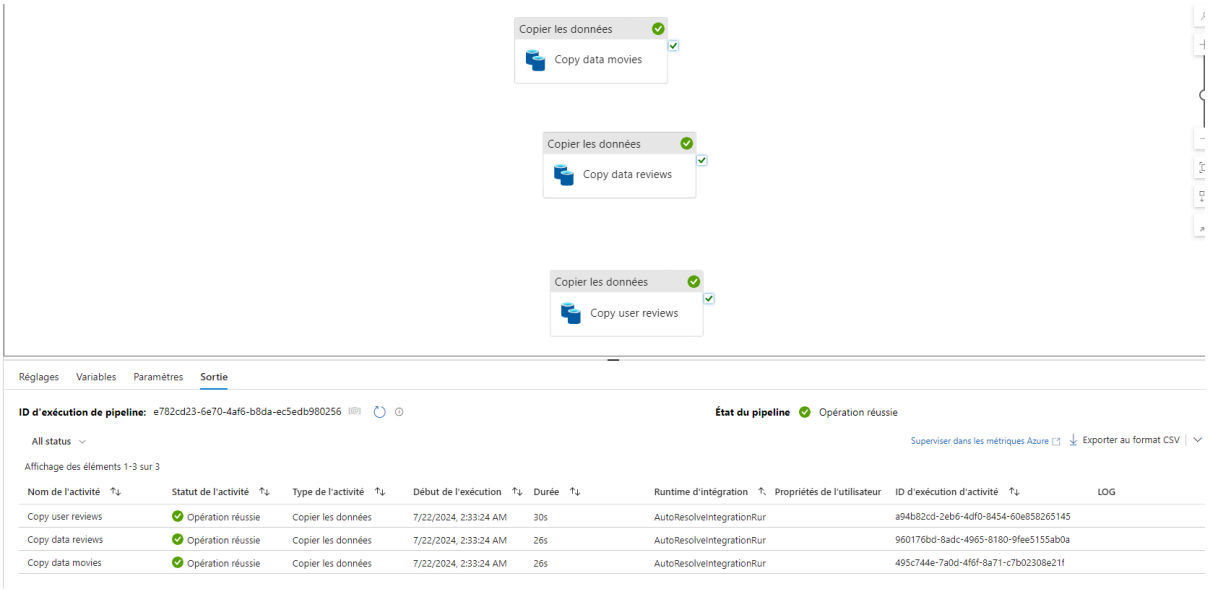
```
-- Dropping existing tables if they exist
IF EXISTS (SELECT * FROM sys.tables WHERE name = 'critics_reviews')
BEGIN
    DROP TABLE [dbo].[critics_reviews];
END;


IF EXISTS (SELECT * FROM sys.tables WHERE name = 'user_reviews')
BEGIN
    DROP TABLE [dbo].[user_reviews];
END;


IF EXISTS (SELECT * FROM sys.tables WHERE name = 'movies')
BEGIN
    DROP TABLE [dbo].[movies];
END;


-- Creating movies table
CREATE TABLE [dbo].[movies]
(
    movieId NVARCHAR(255) NOT NULL,
    movieTitle NVARCHAR(255),
    movieYear INT,
    movieURL NVARCHAR(255),
    critic_score NVARCHAR(255),
    audience_score NVARCHAR(255),
    CONSTRAINT PK_movies PRIMARY KEY NONCLUSTERED (movieId) NOT
ENFORCED
)
WITH
(
    DISTRIBUTION = HASH (movieId),
    CLUSTERED COLUMNSTORE INDEX
);


-- Creating critics_reviews table
```

```
CREATE TABLE [dbo].[critics_reviews]
(
    reviewId NVARCHAR(255),
    movieId NVARCHAR(255), -- Logical relationship with movies table
    creationDate DATE,
    criticName NVARCHAR(255),
    criticPageUrl NVARCHAR(255),
    reviewState NVARCHAR(50),
    isFresh BIT,
    isRotten BIT,
    isRtUrl BIT,
    isTopCritic BIT,
    publicationUrl NVARCHAR(255),
    publicationName NVARCHAR(255),
    reviewUrl NVARCHAR(255),
    quote NVARCHAR(4000),
    scoreSentiment NVARCHAR(255),
    originalScore NVARCHAR(255),
    CONSTRAINT PK_critics_reviews PRIMARY KEY NONCLUSTERED (reviewId) NOT
ENFORCED
)
WITH
(
    DISTRIBUTION = HASH (reviewId),
    CLUSTERED COLUMNSTORE INDEX
);

-- Creating user_reviews table
CREATE TABLE [dbo].[user_reviews]
(
    reviewId NVARCHAR(255),
    movieId NVARCHAR(255), -- Logical relationship with movies table
    rating DECIMAL(3, 1),
    quote NVARCHAR(4000),
    isVerified BIT,
    isSuperReviewer BIT,
    hasSpoilers BIT,
    hasProfanity BIT,
    score FLOAT,
    creationDate DATE,
    userDisplayName NVARCHAR(255),
    userRealm NVARCHAR(255),
    userId NVARCHAR(255),
    CONSTRAINT PK_user_reviews PRIMARY KEY NONCLUSTERED (reviewId) NOT
ENFORCED
)
WITH
(
```

```
    DISTRIBUTION = HASH (movieId),
    CLUSTERED COLUMNSTORE INDEX
);
```

2. **Importation des Données**

Nous avons utilisé Azure Data Factory pour créer des pipelines qui importent les données des fichiers CSV dans les tables Azure Synapse Analytics.



5. Visualisation des Données avec Power BI

1. **Connexion à Azure Synapse Analytics**

Nous avons connecté Power BI à notre serveur Azure Synapse Analytics en utilisant les informations d'identification appropriées.

2. **Création des KPI**

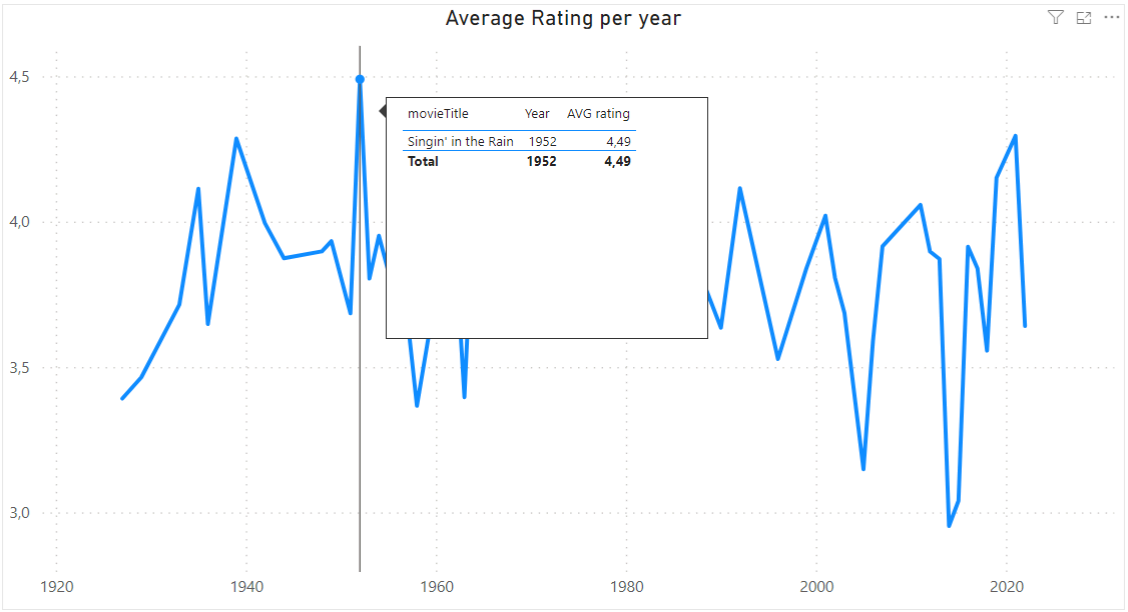
Nous avons créé deux KPI principaux dans Power BI :

- **Les comédies musicales les mieux notées** : Un KPI montrant les films les mieux notés sur la base des scores des critiques et des audiences.

Top Rate film

Movie	Year	Rate
Singin' in the Rain	1952	4,49
Tick, Tick... Boom!	2021	4,42
Sing Street	2016	4,41
Lagaan: Once Upon a Time in India	2001	4,39
In the Heights	2021	4,34
Fiddler on the Roof	1971	4,30
The Wizard of Oz	1939	4,29
The Blues Brothers	1980	4,28
West Side Story	3982	4,28
The Court Jester	1955	4,25
Blinded by the Light	2019	4,25
Hedwig and the Angry Inch	2001	4,14
My Fair Lady	1964	4,14
Rocketman	2019	4,14
Once	2007	4,13
A Hard Day's Night	1964	4,13
Top Hat	1935	4,13
Mary Poppins	1964	4,12
Hipsters	2011	4,12
French Cancan	1955	4,12
The Muppet Movie	1979	4,12
The Muppet Christmas Carol	1992	4,12
A Night at the Opera	1935	4,10
The Sound of Music	1965	4,06
The Muppets	2011	4,06
The Umbrellas of Cherbourg	1964	4,06
White Christmas	1954	4,06

- **Évolution de la Note Globale Moyenne** : Un KPI montrant l'évolution des notes globales moyennes des comédies musicales par année.



6. Résultats et Insights

Les visualisations Power BI nous ont permis d'identifier les comédies musicales les mieux notées et de suivre les tendances des notes au fil des ans. Cela peut aider à comprendre l'évolution de la popularité et de la qualité perçue des comédies musicales.

Conclusion

Le projet a permis de mettre en place une infrastructure complète pour le nettoyage, le stockage, l'analyse et la visualisation des données des critiques de comédies musicales. Les insights obtenus peuvent être utilisés pour des analyses plus approfondies et pour guider les décisions dans l'industrie du divertissement.

Si vous avez des questions supplémentaires ou avez besoin de plus de détails sur certaines parties du projet, n'hésitez pas à demander.