

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from the bar, containing the date.

10/9/2023

P2 Design – ADBMS Team 6

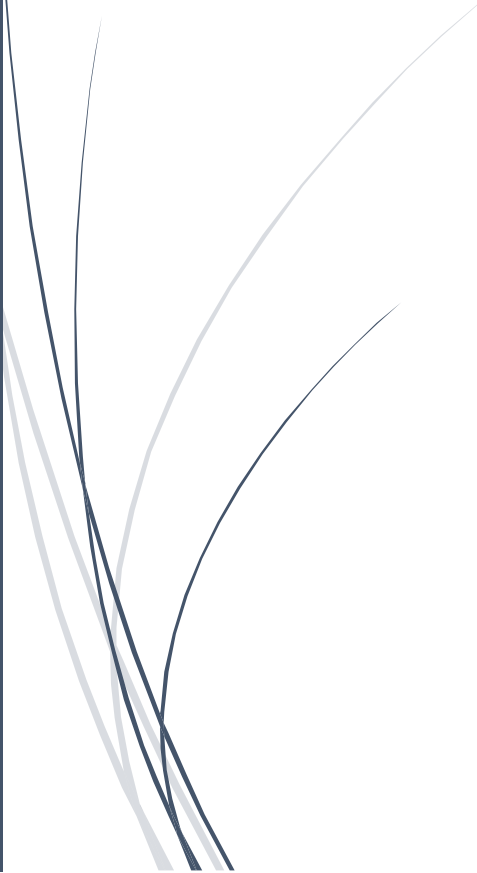


Table of Contents

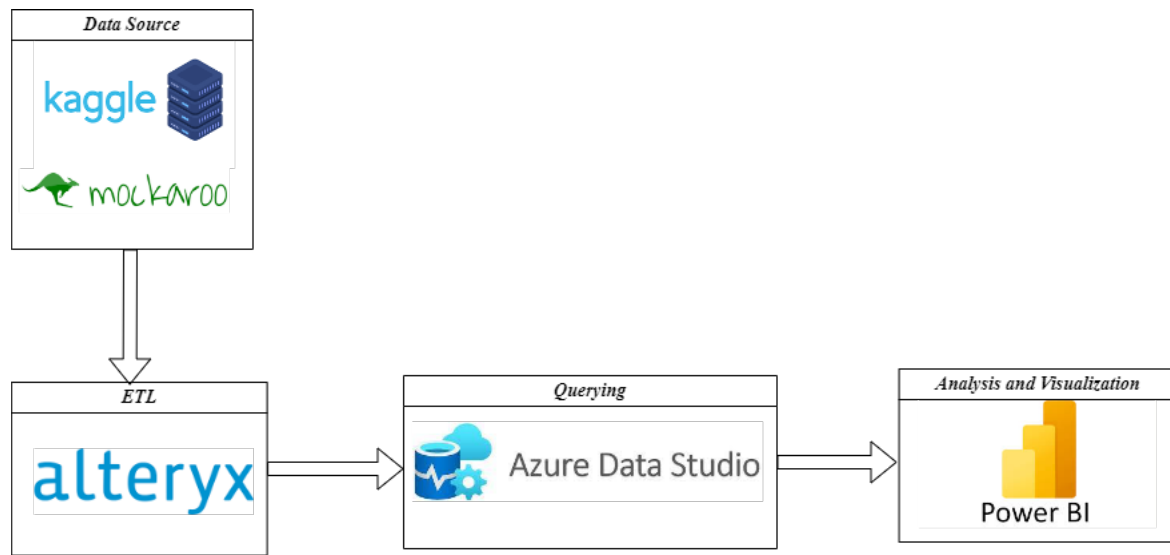
1. ARCHITECTURE DIAGRAM..... 2

1. RELATIONAL DIAGRAM..... 3

2. GRAPH DATA MODEL..... 4

3. DOCUMENT DATA MODEL 5

1. Architecture Diagram



The architectural workflow is designed to efficiently manage data from multiple sources and transform it into actionable insights. Here's a breakdown of the process:

Data Acquisition (Kaggle and Mockaroo): The journey begins by sourcing data from Kaggle and Mockaroo. These reputable data platforms provide the foundation for our analytical work, akin to acquiring reference material for an academic study.

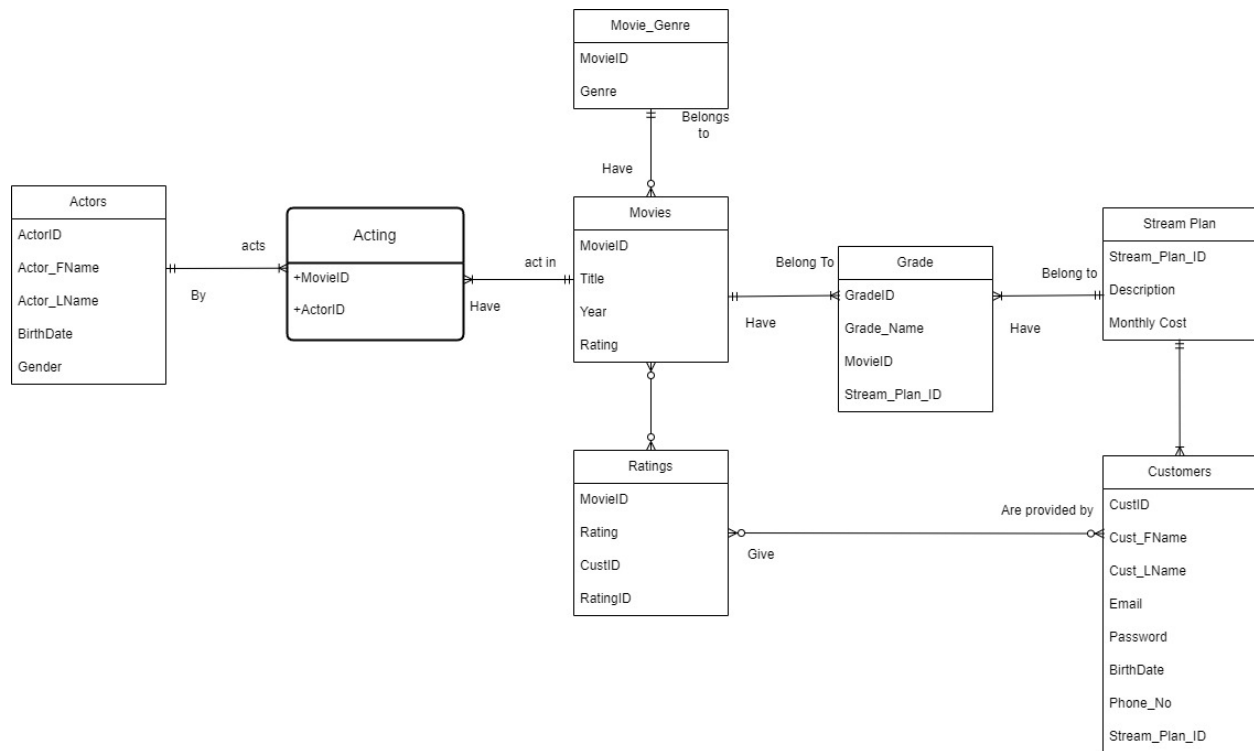
Data Transformation (Alteryx - ETL): The acquired data often arrives in varying states of cleanliness, similar to academic research data that requires organization. Alteryx serves as our Extract, Transform, Load (ETL) tool, ensuring data integrity and consistency.

Data Storage (Azure): Azure functions as our secure repository, akin to a well-structured library archive. It houses the cleaned and organized data, making it easily accessible while maintaining data security.

Data Query and Analysis (Azure): Utilizing Azure, we perform data queries, resembling research inquiries. Azure's capabilities allow us to swiftly retrieve specific information from our stored data, equivalent to conducting targeted research within a vast knowledge base.

Data Visualization (Power BI): Power BI plays a pivotal role in turning data into comprehensible insights, similar to presenting research findings in a visually engaging manner. It assists in conveying complex information through clear and insightful visualization.

1. Relational Diagram



The relational ERD contains the following tables:

Movie: This table stores information about the movies, such as their title, genre, release date, and rating.

Genre: This table stores information about the genres of movies, such as action, comedy, drama, and horror.

Ratings: This table stores information about the ratings of movies, such as the average rating and the number of ratings.

Acting: This table is an associative entity that stores information about the relationship between movies and actors.

Actors: This table stores information about the actors, such as their name, age, and gender.

Grade: This is an associative table that also stores information about the grades of stream plans, such as standard, premium, and ultra-HD.

Stream Plan: This table stores information about the stream plans, such as the monthly cost and the number of screens allowed.

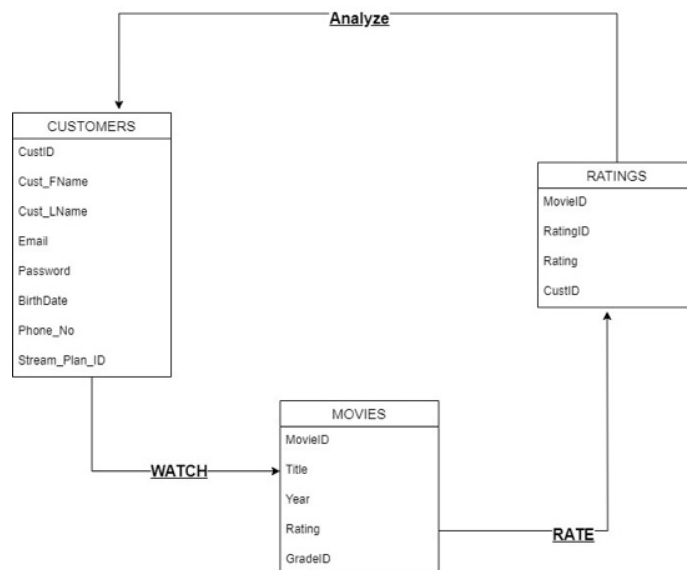
Customers: This table stores information about the customers, such as their name, email address, and phone number.

The tables are related to each other as follows:

A movie can have one or more genres.
A movie can have one or more actors.
An actor can appear in one or more movies.
A stream plan can be subscribed to by one or more customers.

This ERD can be used to store and manage data about a streaming service. For example, the database could be used to generate reports on the most popular movies, the most popular actors, and the most popular stream plans. The database could also be used to recommend movies to customers based on their viewing history.

2. Graph Data Model



WATCHED: User -> Movie

This relationship shows when users have watched movies. It's like connecting users and the movies they've seen.

Each time a user watches a movie, we create a "WATCHED" link between that user and the movie.

For example, if User A watches Movie X, there's a "WATCHED" link between User A and Movie X.

RATED: User -> Movie -> Rating

This relationship goes a step further and is about users giving ratings to movies.

It connects users to movies they've watched and adds ratings to the mix.

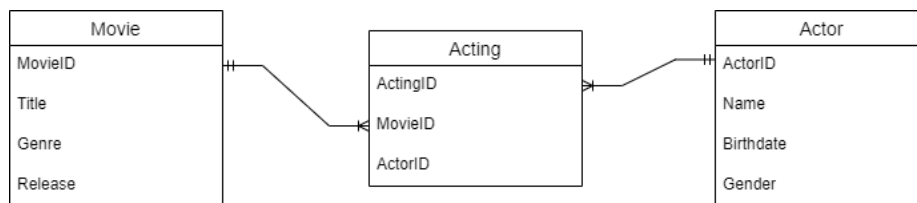
Every time a user rates a movie, we create a "RATED" link between the user, the movie, and the rating.

For instance, if User B watches Movie Y and gives it 4 out of 5 stars, there's a "RATED" link connecting User B, Movie Y, and the rating.

We use this relationship to understand user preferences better by looking at how they rate different movies.

In summary, these relationships help us keep track of which movies users watch and how they rate them. "WATCHED" shows user-movie interactions, while "RATED" adds extra detail by recording user ratings. This data is crucial for understanding what users like and for finding popular movies based on their ratings.

3. Document Data Model



In Azure Data Studio, we've designed a document-like model for Movies and Actors using structured tables. Here's the key structure:

Movies Table: Stores movie details (title, genre, release date).

Actors Table: Stores actor details (name, birthdate).

Acting Table: Links movies and actors together, allowing us to connect actors to the movies they've appeared in.

This approach allows us to work with structured data efficiently in Azure Data Studio and perform SQL queries to find actors in specific movies or movies featuring particular actors. It's a suitable choice for scenarios where you need to maintain relationships between entities in a tabular environment.