**Overview**

To support your business objectives, we designed and implemented a SQL database to manage and analyse movie data efficiently. Below is a summary of key operations and decisions, explained in simple, non-technical language.

**1. Database Creation**

We created a new database called streamflixDb. This acts like a digital filing cabinet – it holds all the information about Movies, Users, Genre, and Ratings in one organised space.

**2. Table Structures**

We set up different “tables” (like spreadsheets) for each major category of data. For example:

* **Movies**: Stores titles, release years, languages, and view counts.
* **Genres**: Tracks movie genres (e.g. comedy, drama).
* **MovieGenres**: Tracks movie genres (e.g. comedy, drama).
* **Users**: Views what time, age group, device used, subscription type
* **Ratings**: To analyse use satisfaciotn and get more insights to convert free users to subscription based

Breaking data into logical tables keeps things clean, helps avoid duplication, and speeds up analysis.

**3. Genre Handling**

Instead of cramming multiple genres into one field, we used a separate table to handle them properly. Each movie can link to multiple genres. This makes it easier to count how many movies fall under each genre and compare performance across types.

**4. Data Import**  
We built a python script that loads all your movie data from CSV files into the database.

This saves time and ensures the data is loaded accurately every time – no more copy-pasting.

**6. Quality Checks**

We added checks to catch any missing or incorrect information before it enters the database**.** Good data in = good insights out. This reduces errors and keeps your reports reliable.

**7. Performance Considerations**

We used indexing and structured the data efficiently to keep things fast and responsive. Your reports and dashboards will load quickly, even as your data grows.

**Movies**

| **Column Name** | **Data Type** | **Key** | **Notes** |
| --- | --- | --- | --- |
| MovieID | INT | Primary Key | Unique movie ID |
| Title | NVARCHAR(255) |  | Movie title |
| Year | NVARCHAR(4) |  | Release year |
| Language | NVARCHAR(100) |  | Movie language |
| Country | NVARCHAR(100) |  | Country of origin |
| TotalViews | INT |  | Total views count |

**Genres**

| **Column Name** | **Data Type** | **Key** | **Notes** |
| --- | --- | --- | --- |
| GenreID | INT (IDENTITY) | Primary Key | Auto-incremented ID |
| GenreName | NVARCHAR(100) | Unique | Genre name (e.g. Drama) |

**MovieGenres (Join Table)**

| **Column Name** | **Data Type** | **Key** | **Notes** |
| --- | --- | --- | --- |
| MovieID | INT | PK, FK | Refers to Movies.MovieID |
| GenreID | INT | PK, FK | Refers to Genres.GenreID |

Composite Primary Key: (MovieID, GenreID)

**Users**

| **Column Name** | **Data Type** | **Key** | **Notes** |
| --- | --- | --- | --- |
| UserID | NVARCHAR(20) | Primary Key | Unique user ID |
| Age | INT |  | Age of user |
| Gender | CHAR(1) |  | M / F |
| Country | NVARCHAR(100) |  | User country |
| SubscriptionStatus | NVARCHAR(50) |  | e.g., Active, Cancelled |
| TotalWatchTime | INT |  | Total minutes watched |
| Device | NVARCHAR(50) |  | e.g., Mobile, Desktop |

**Ratings**

| **Column Name** | **Data Type** | **Key** | **Notes** |
| --- | --- | --- | --- |
| RatingID | INT | Primary Key | Unique rating ID |
| UserID | NVARCHAR(20) | FK | Refers to Users.UserID |
| MovieID | INT | FK | Refers to Movies.MovieID |
| Rating | DECIMAL(2,1) |  | e.g., 4.5 |
| Timestamp | DATETIME |  | When the rating was submitted |