**System Documentation   
  
SQL Queries Documentation**

Each query serves a specific purpose, ranging from analyzing seasonal movie trends to exploring streaming platform statistics. The database schema supports these queries through a robust design featuring relational tables with key associations. This documentation outlines the purpose of each query, its SQL implementation, and how the database structure supports its execution.

Query 1: Seasonal Trends in Movie Genres

Purpose: This query identifies the top five movie genres in a given season and year, based on average popularity and total revenue.

Database Design Support:

* The *Movies* table provides release dates and metadata.
* The *MovieGenres* table links movies to their genres.
* The *MovieFinances* table captures revenue data.
* The database stores movies, genres, and ratings in separate tables linked by foreign keys, enabling efficient join operations.
* For user inputs like "Action", full-text indexing on the genre table allows for efficient lookups.

Query 2: Frequent Co-Actors

Purpose: This query identifies the top two actors who frequently co-star with a given primary actor, providing insights into actor collaborations.

Database Design Support:

* The *MovieActors* table links actors to movies.
* The *Actors* table contains actor names and IDs.
* Relationships between these tables facilitate the identification of co-actors.

Query 3: Movie Search by Name

Purpose: This query searches for movies by name, returning their details and available streaming platforms, sorted by rank.

Database Design Support:

* The *Movies* table supports full-text search on titles.
* The *MovieProviders* table connects movies to streaming platforms.
* Indexing ensures efficient searching and grouping operations.

Query 4: Top Streaming Platform for Highly Ranked Movies

Purpose: This query determines the streaming platform with the highest number of unique movies ranked above a specified threshold.

Database Design Support:

* The *Movies* table provides ranking data.
* The *MovieProviders* table tracks movie availability across platforms.
* Filtering and grouping are optimized through indexing.
* Linking *movies* to *genres* enables straightforward revenue calculations via group-by queries.

Query 5: Advanced Movie Search

Purpose: This query performs an advanced search based on free text, genre, release date, and runtime, returning detailed movie information.

Database Design Support:

* The *Movies* table supports full-text search and provides runtime and release dates.
* The *MovieGenres* and *MoviesFinance* tables supply genre and financial data.
* Full-text indices enable efficient free-text searches.

These queries demonstrate the capabilities of the movie database to provide actionable insights and support user requirements. The relational schema and indexing strategies ensure efficient query execution, enabling diverse use cases from trend analysis to advanced searches.