Software Documentation

**Database Scheme:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table Keys** | | **Table columns** | | **Table name** |
| **FK** | **PK** | **Type** | **Name** |
| None | id | INT | id | movie |
| VARCHAR(256) | title |
| DATE | release\_date |
| INT | runtime |
| VARCHAR(1024) | description |
| FLOAT | rating |
| INT | production\_budget |
| INT | marketing\_budget |
| INT | revenue |
| None | id | INT | id | person |
| VARCHAR(256) | full\_name |
| DATE | birth\_date |
| DATE | death\_date |
| None | id | INT | id | genre |
| VARCHAR(128) | title |
| None | id | INT | id | role |
| VARCHAR(128) | title |
| * movie\_id   REFERNCES  movie(id)   * genre\_id   REFERENCES genre(id) | None | INT | movie\_id | movie\_genre |
| INT | genre\_id |
| * movie\_id   REFERNCES  movie(id)   * person\_id   REFERENCES person(id)   * role\_id   REFERENCES role(id) | None | INT | movie\_id | movie\_role |
| INT | person\_id |
| INT | role\_id |

**Database Indexes (excluding keys):**

* Index on movie.release\_date
* Index on movie.rating
* Full-text index on movie.description
* Full-text index on person.full\_name

**Database Queries:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Query**  **Name** | **Main/**  **Secondary** | **Description** | **Parameters** |
| query\_1 | Main | Return a list of persons who had a better than average rating in a specific genre while performing the given role. List ordered by a person's average rating for projects fitting role and genre, in descending order. | * role * genre |
| query\_2 | Main | Computes the total revenue for each movie genre since the specified start year (inclusive). Return a list ordered by revenue descending. | * starting release year |
| query\_3 | Main | Retrieves the top 100 most active people in a specific role since a given starting release year, ordered by person count of projects. | * role * starting release year |
| query\_4 | Main | Retrieves people with a specific string in their full name. | * search string |
| query\_5 | Main | Retrieves movies which have specific strings (up to 3 strings) in their description (AND logic). | * search string 1 * search string 2 * search string 3 |
| query\_6 | Secondary | Retrieves all roles. | None |
| query\_7 | Secondary | Retrieves all genres. | None |

**Database optimization:**

1. Chosen scheme design explanation:

ddd

1. Index optimization explanation:

ddd

**API Usage:**

We used the IMDB public databases in form of .csv files. We also added information from different sources about revenue and movie description to increase options for more diverse queries. We use 'pandas' python module to update our table information from those .csv files.

**Application workflow:**

Application 'MovieRanger' is an application designed to streamline the process for film studios when planning and producing new project by providing data on the latest successful movies and working talent in the industry. When opening the application, the user already sees information about the most successful genres in the latest year, to help him decide what genre to focus on in the new project. Next, there are multiple page options for planning: Search for active/highly rated talent in the industry to hire; Look up specific person of interest; Search movies by keywords for more specific themes. More information in the user manual…