CS306 Database Management

Project Report: Movie Database

Almir Bajric

Ammar Haljkovic

Faruk Basic

1. Description:

In developing a comprehensive movie database, the primary challenge was to efficiently organize and manage the diverse and multifaceted information associated with movies. This information encompasses various aspects such as genres, production details, actor profiles, user ratings, and the intricate relationships among these entities. The initial step involved identifying the key entities relevant to the movie database. These entities included movies, categories, actors, production companies, and user ratings, each with their own set of attributes that needed to be accurately captured.

Following this, the design phase commenced with the creation of an Enhanced Entity-Relationship (EER) diagram. This diagram was crucial as it provided a visual representation of the entities and the relationships between them, highlighting the cardinalities and constraints. The EER diagram helped in understanding how different data points interacted and how to structure the database to accommodate these interactions efficiently.

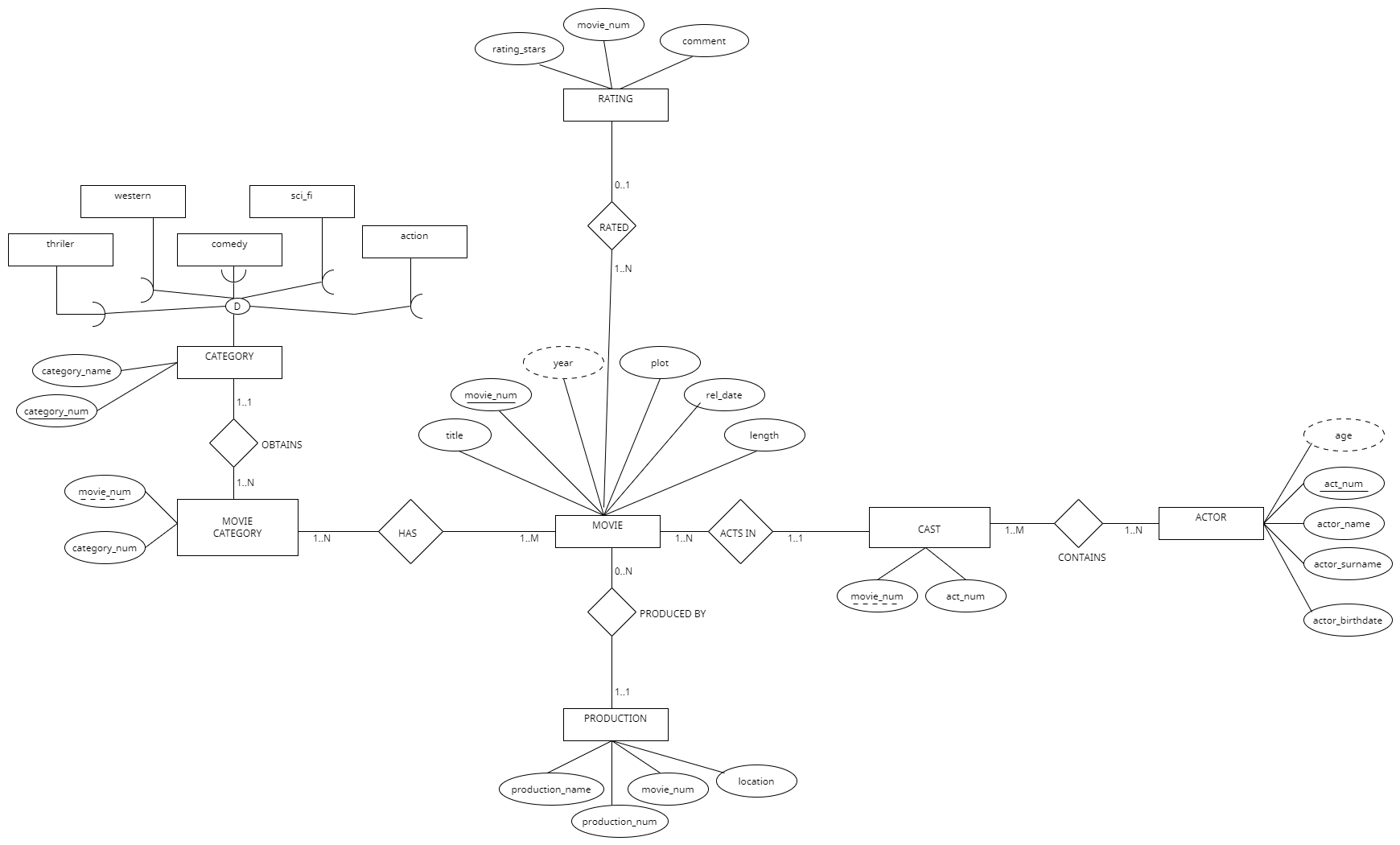
The next phase involved translating the EER diagram into a relational schema. This step was essential to ensure that the database was normalized, reducing data redundancy and ensuring data integrity. Primary keys were defined for each table to uniquely identify records, while foreign keys were used to establish relationships between tables, ensuring referential integrity. This structured approach allowed for the development of a database capable of supporting complex queries and operations, facilitating robust data management and retrieval.

**2. EER Diagram:**

The EER diagram provides a detailed schema for managing a movie database, encompassing entities such as CATEGORY, MOVIE, ACTOR, PRODUCTION, and RATING, along with their relationships. The CATEGORY entity lists genres like western, sci-fi, comedy, thriller, and action, each identified by a unique category number and name. These categories are linked to the MOVIE CATEGORY entity, which associates specific movies with their respective genres through relationships that indicate a category can encompass multiple movie categories, and each movie category belongs to one category. The MOVIE entity is central, containing attributes like movie number, title, year of release, plot summary, release date, and length. It connects to multiple other entities: it is linked to the MOVIE CATEGORY entity, signifying that each movie can fall under multiple categories.

The ACTOR entity, characterized by attributes such as actor number, first name, surname, birthdate, and optionally age, is associated with the CAST entity, which links actors to the movies they have acted in. This relationship shows that a movie can have multiple actors, and each cast entry involves one movie. MOVIE also connects to the RATING entity, which allows users to rate movies with stars and comments, indicating a movie can have multiple ratings, but each rating pertains to one movie. Furthermore, the MOVIE entity is linked to the PRODUCTION entity, which includes details like production number, name, and location of the production company, showing that each movie is produced by one company, while a production company can produce multiple movies.

The relationships defined in the diagram, such as OBTAINS (linking CATEGORY and MOVIE CATEGORY), HAS (linking MOVIE and MOVIE CATEGORY), RATED (linking MOVIE and RATING), ACTS IN (linking MOVIE and CAST), CONTAINS (linking CAST and ACTOR), and PRODUCED BY (linking MOVIE and PRODUCTION), create a comprehensive and interconnected structure. This structure ensures detailed and organized data management, allowing for a robust database that can effectively handle the various aspects of movie information, from genre categorization and actor participation to user ratings and production details.



**3. Relational scheme:**

This is our relational database schema for a movie database, comprising several interconnected tables to manage various aspects of movie information. The CATEGORY table contains fields for category numbers and names, allowing classification of movies into different genres. The MOVIE table is more comprehensive, with fields for movie numbers, titles, years, plots, release dates, lengths, and production numbers, linking each movie to a specific production. The ACTOR table includes actor numbers, names, surnames, birthdates, and ages, capturing detailed information about each actor. The PRODUCTION table records production numbers, names, and locations, identifying the companies or entities responsible for producing the movies. The RATING table collects ratings and comments for each movie, with fields for rating IDs, movie numbers, stars, and comments, enabling user feedback and evaluation. The schema also includes two associative tables to manage many-to-many relationships: the MOVIE\_CATEGORY table links movies to categories via movie numbers and category numbers, and the CAST table connects movies to actors using movie numbers and actor numbers. This schema effectively organizes and relates all necessary data to support a robust movie database system.

**The red represents primary keys while the light blue represents foreign keys.**

CATEGORY(category\_num, category\_name)

MOVIE(movie\_num, title, year, plot, rel\_date, length, production\_num)

ACTOR(act\_num, actor\_name, actor\_surname, actor\_birthdate, age)

PRODUCTION(production\_num, production\_name, location)

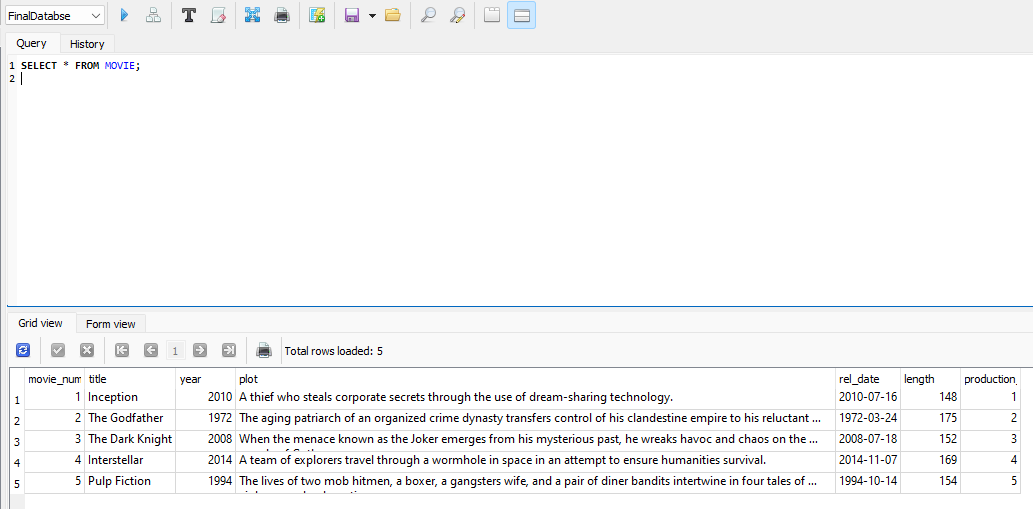
RATING(rating\_id, movie\_num, rating\_stars, comment)

MOVIE\_CATEGORY(movie\_num, category\_num)

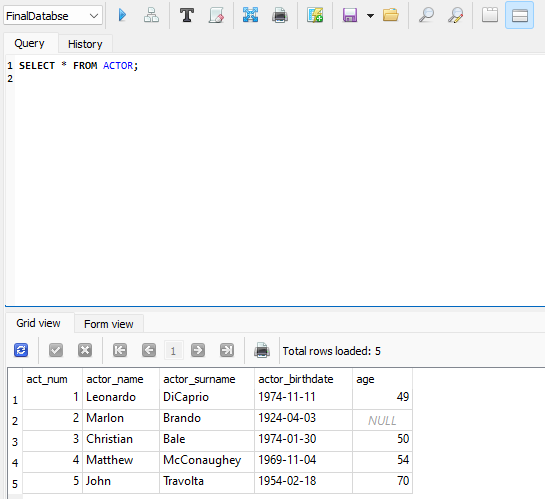
CAST(movie\_num, act\_num)

**5. Questions**

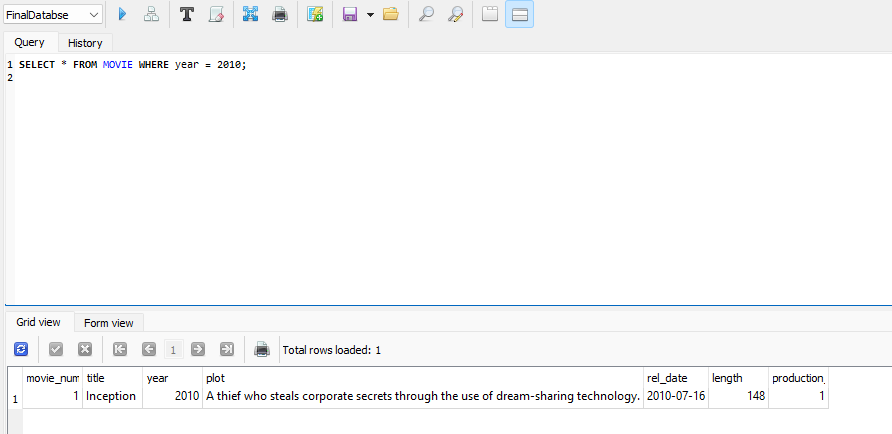
1. List all movies in the database.



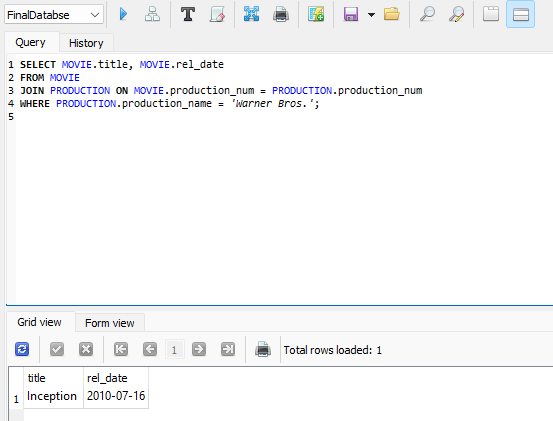
2. List all actors in the database.



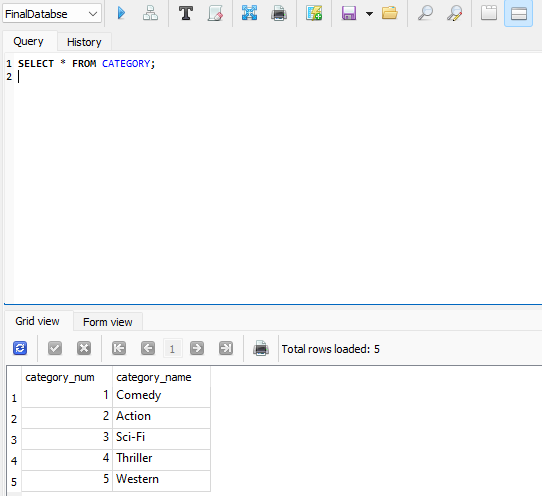
3. Find all movies released in a specific year, e.g., 2010.



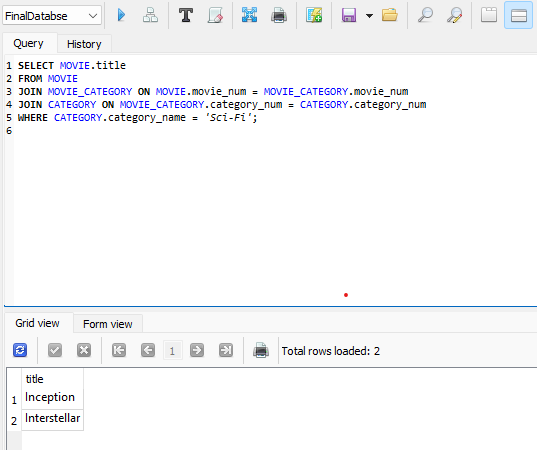
4. Get the title and release date of all movies produced by a specific production company, e.g., Warner Bros.



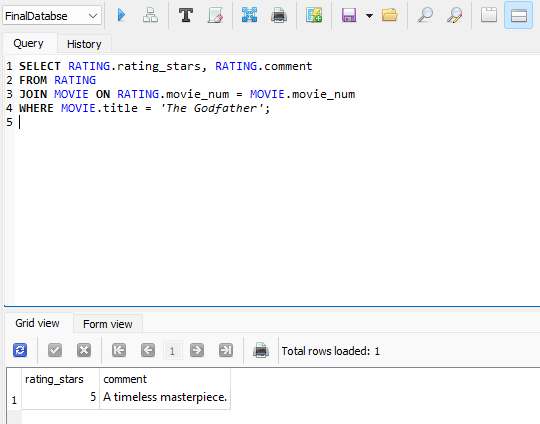
5. List all categories available in the database.



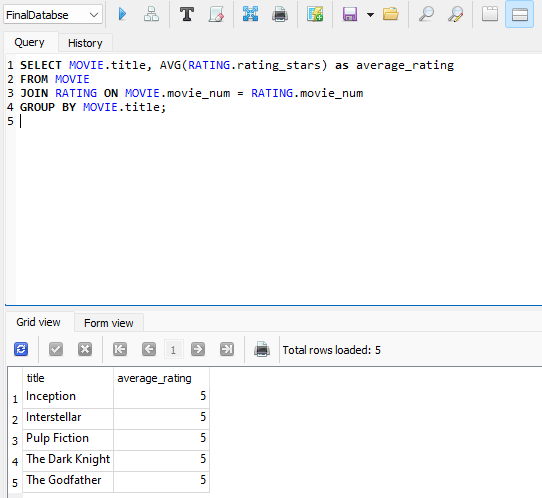
6. Find all movies that fall under the 'Sci-Fi' category.



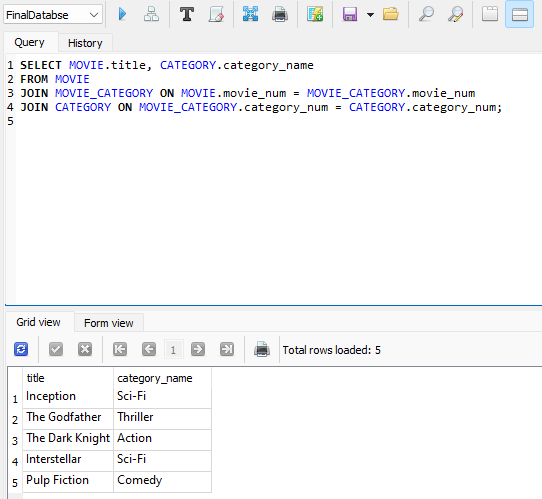
7. List all ratings and comments for a specific movie, e.g., 'The Godfather'.



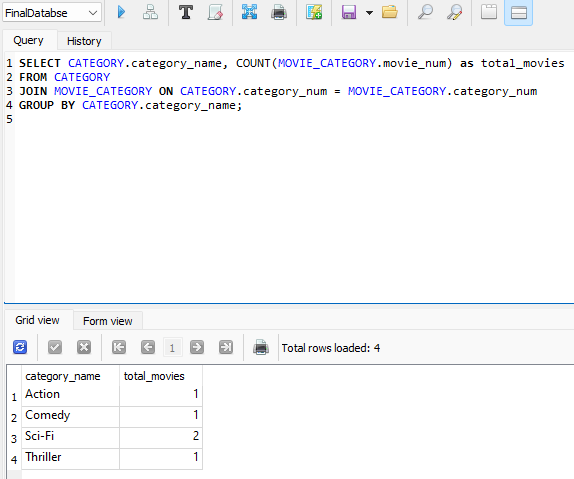
8. Find the average rating of each movie.



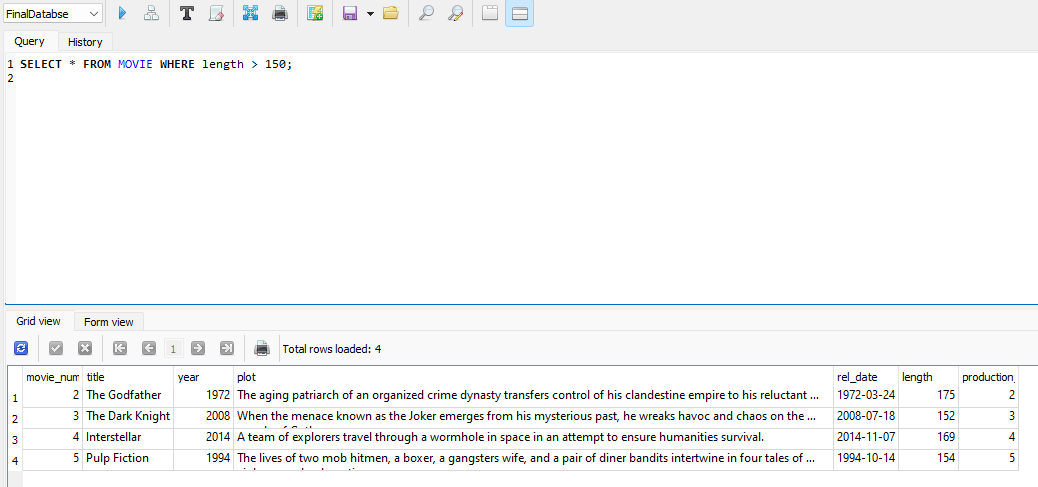
9. List all movies along with their categories.



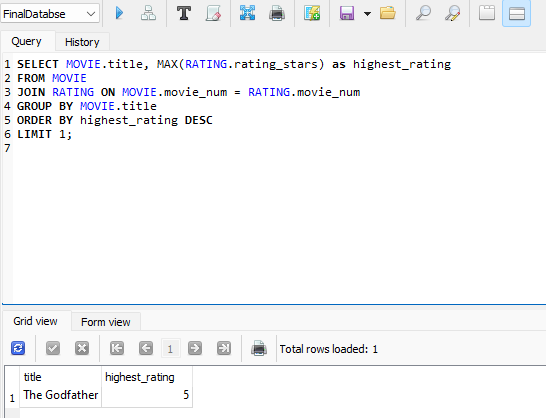
10. Get the total number of movies in each category.



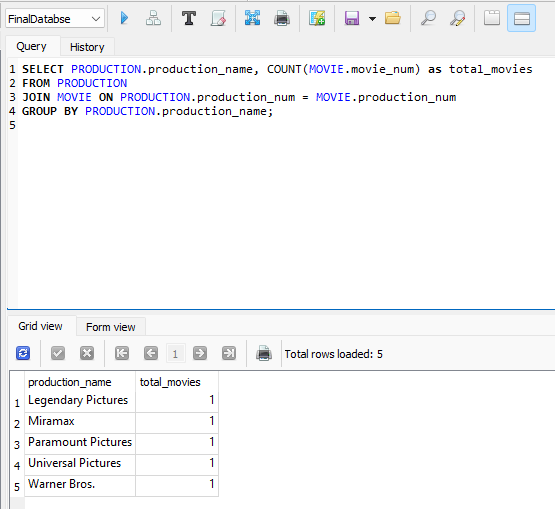
11. Find all movies longer than a specific length, e.g., 150 minutes.



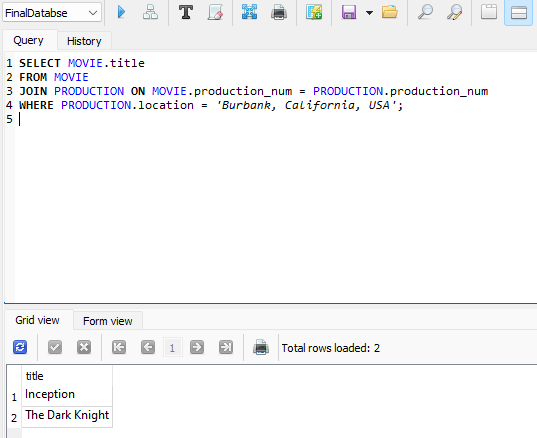
12. Get the details of the movie with the highest rating.



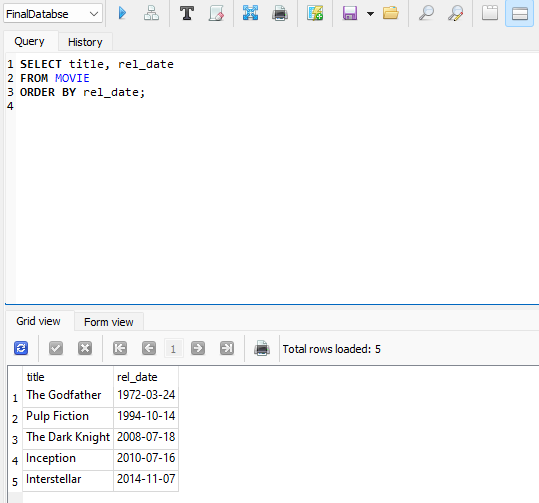
13.List all production companies along with the number of movies they have produced.



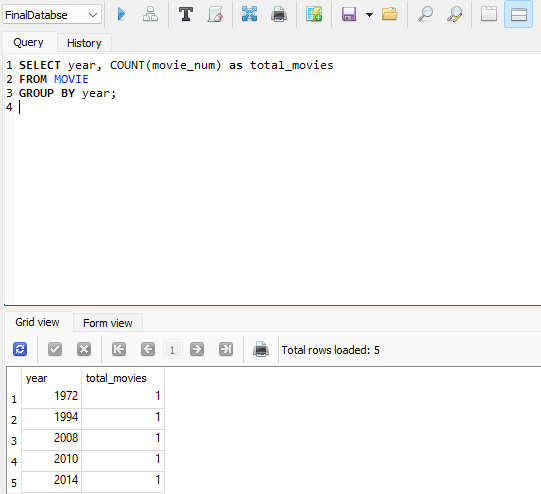
14. Find all movies produced in a specific location, e.g., 'Burbank, California, USA'.



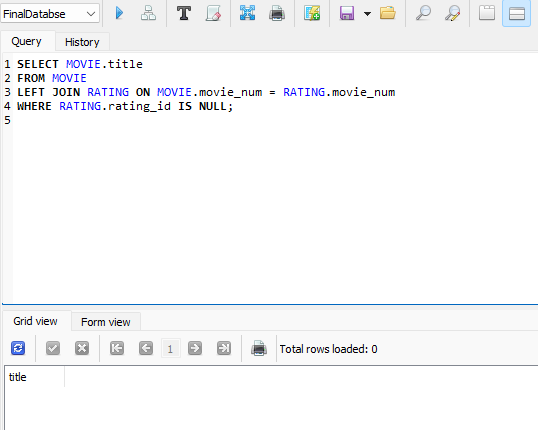
15. Get the list of movies and their release dates sorted by the release date.



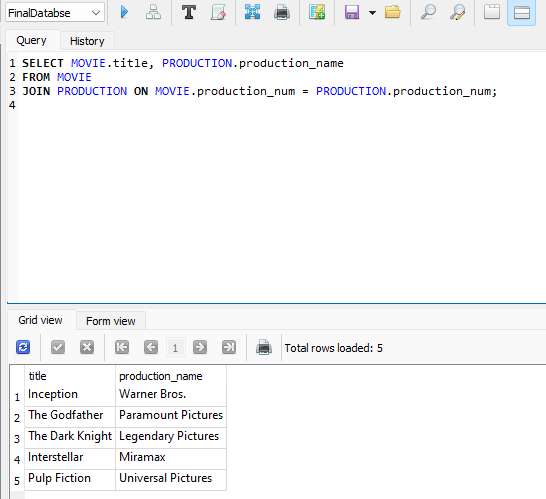
16. Find the number of movies released each year.



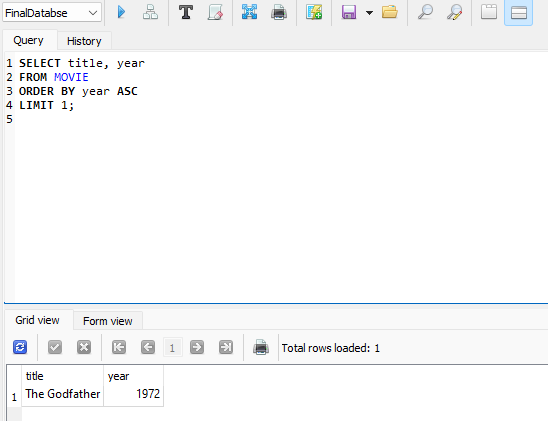
17. Find all movies that do not have any ratings.



18. List the titles of movies along with their production company names.



19. Find the oldest movie in the database.



20. List the titles and plots of movies with a specific keyword in the plot, e.g., 'space'.

