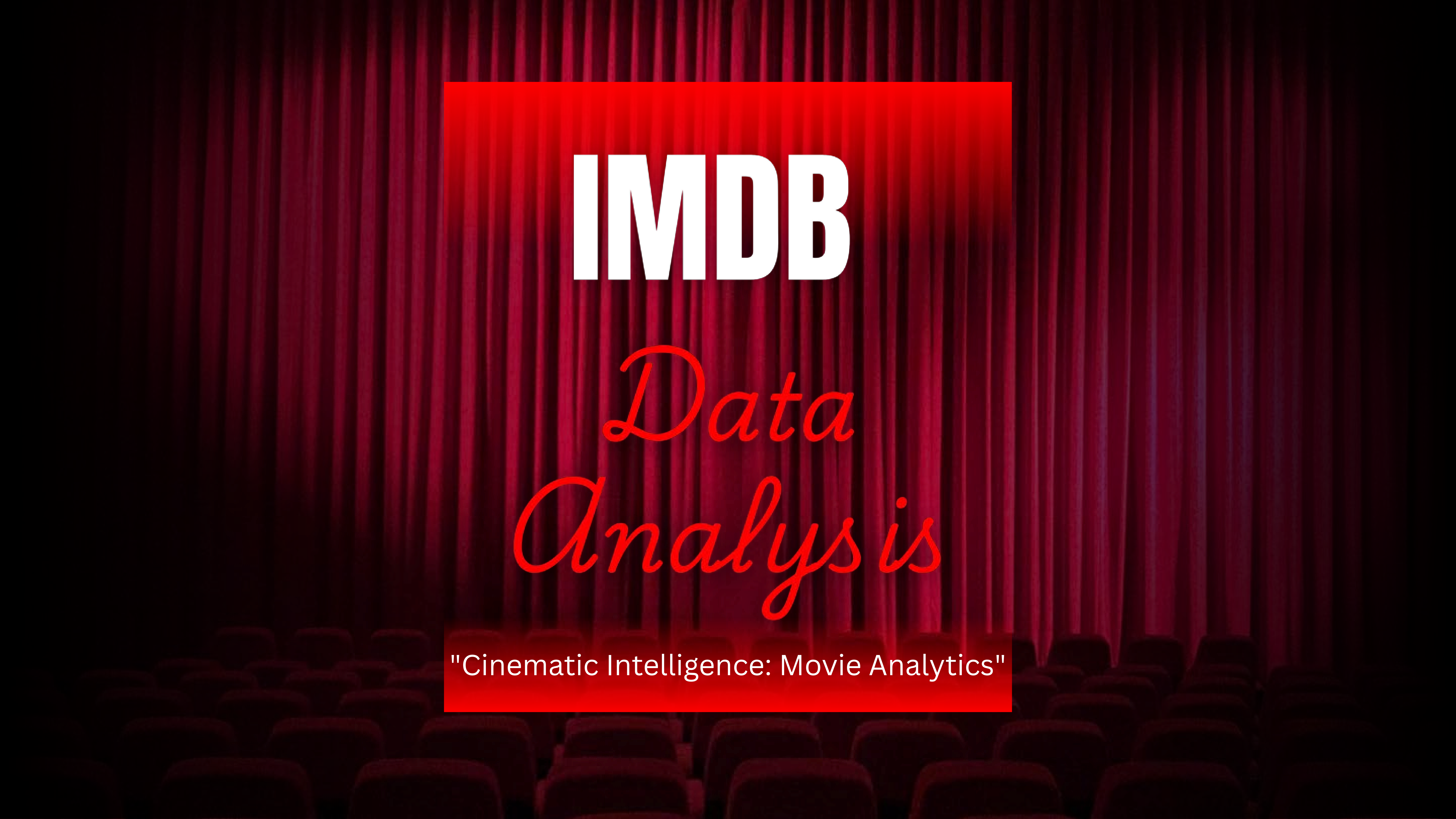
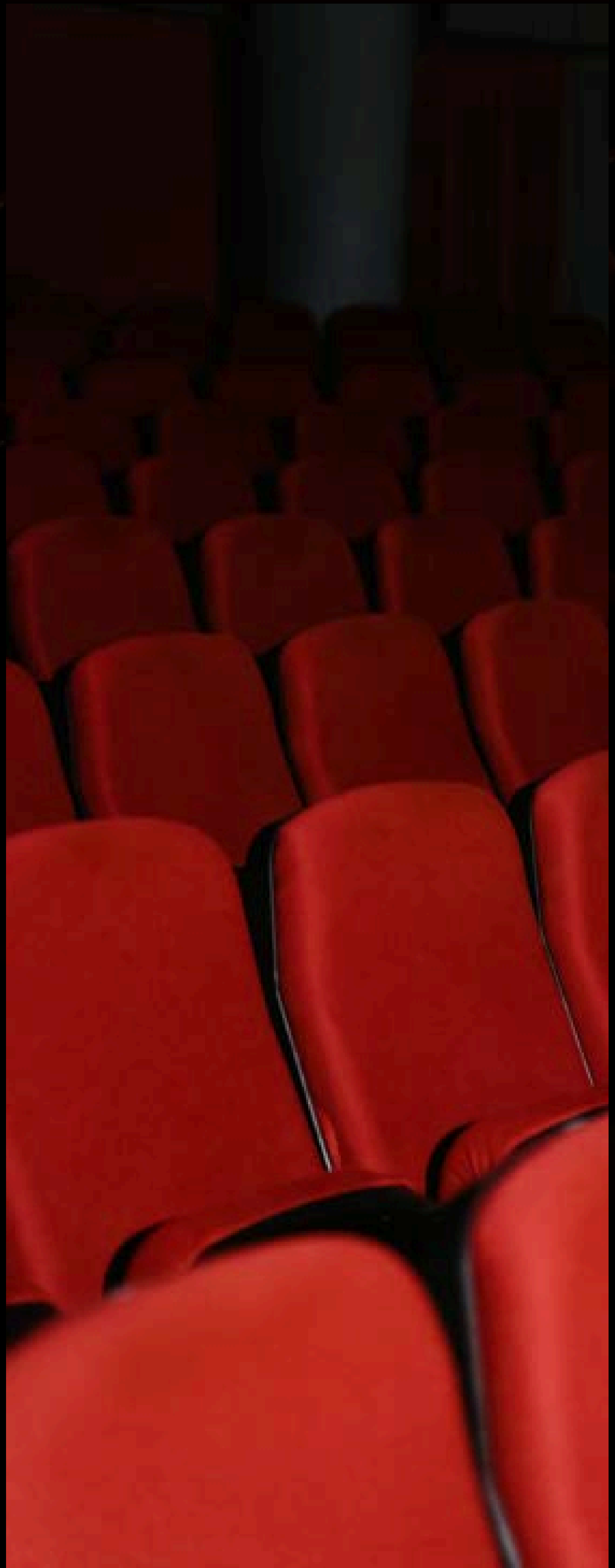


IMDB

Data Analysis

"Cinematic Intelligence: Movie Analytics"



Content Table

- Introduction
- Dataset Description
- Objectives & Scope
- Workflow & Approach
- Data Collection & Loading
- Data Cleaning
- Data Analysis
- Insights & Findings
- Business Impact

Introduction

- This project focuses on analyzing IMDB movie data to uncover audience preferences, rating patterns, and content performance.
- By utilizing SQL queries on the Movies and Ratings tables, the analysis aims to extract meaningful insights that help identify top-rated movies, genre trends, and viewer sentiments, enabling data-driven decisions in the film industry.

Dataset Description

- The IMDB movie dataset, sourced from Kaggle, provides comprehensive information for analyzing movies and audience ratings.

- It consists of two tables — Movies and Ratings. The Movies table includes details such as movie titles, genres, release dates, durations, and languages, while the Ratings table contains user ratings and reviews.

- Linked via movie id, this dataset enables in-depth analysis of audience preferences, rating patterns, and content performance.

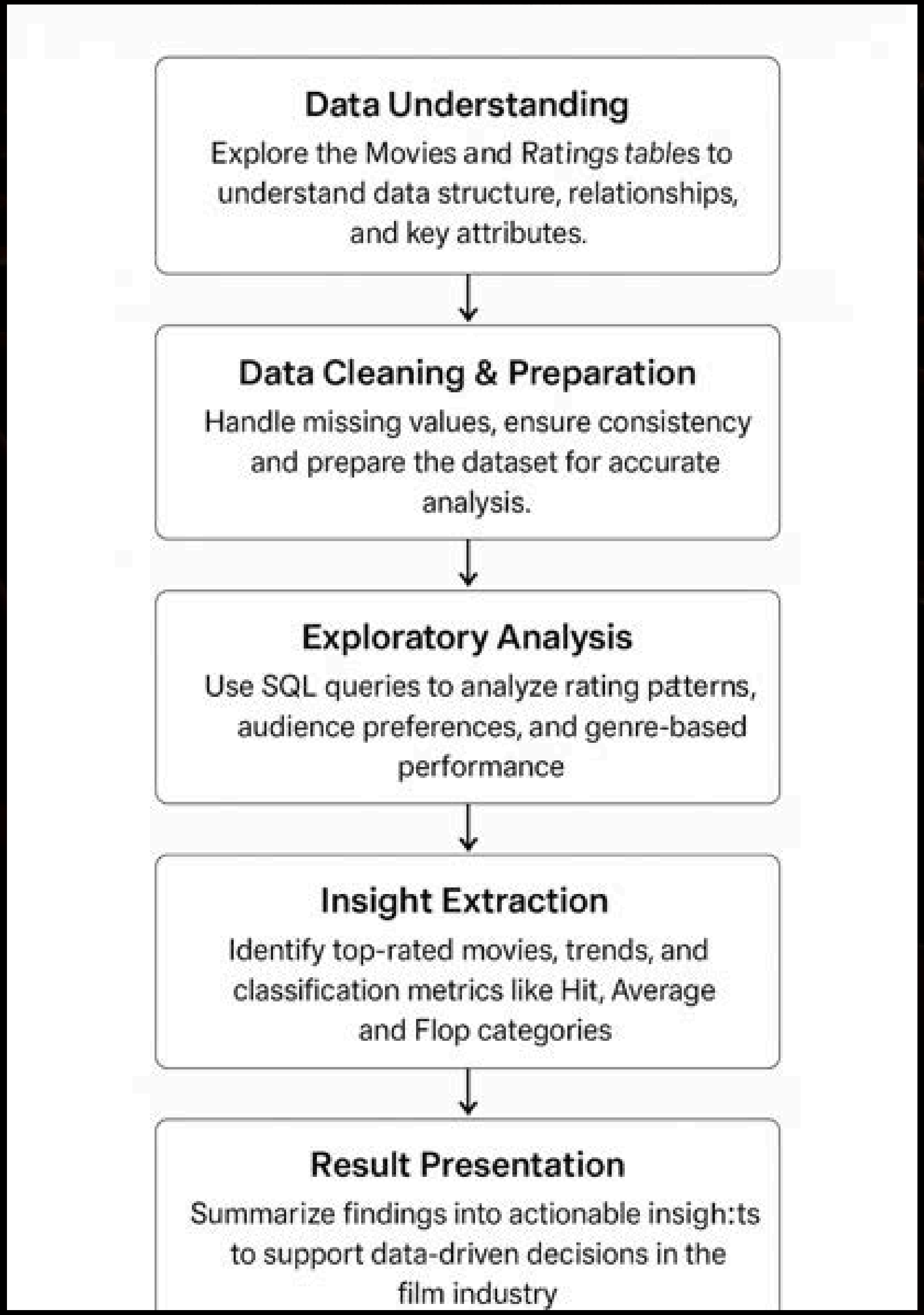
Objective

- Ensure Data Accuracy & Consistency – Maintain high-quality movie and ratings data to enable reliable analytics.
- Uncover Audience Insights – Analyze ratings and reviews to understand viewer preferences, trends, and genre performance.
- Empower Data-Driven Decisions – Provide actionable insights for optimizing content strategies and enhancing audience engagement.

Scope

- Analyze IMDB movie and ratings data to uncover audience preferences, evaluate content performance, optimize insights for operational efficiency, and support data-driven decision-making.

Workflow & Approach



Data Understanding

Explore the Movies and Ratings *tables* to understand data structure, relationships, and key attributes.

↓

Data Cleaning & Preparation

Handle missing values, ensure consistency and prepare the dataset for accurate analysis.

↓

Exploratory Analysis

Use SQL queries to analyze rating patterns, audience preferences, and genre-based performance

↓

Insight Extraction

Identify top-rated movies, trends, and classification metrics like Hit, Average and Flop categories

↓

Result Presentation

Summarize findings into actionable insigh:ts to support data-driven decisions in the film industry

Data Collection & Loading

Data Sources

IMDB dataset obtained from Kaggle, consisting of two tables:

- Movies Table – Contains movie ID, title, genres, release year, and duration.
- Ratings Table – Includes user ratings, number of reviews, and average ratings.

Data Loading Process

The dataset was directly imported into MySQL Workbench using the Table Data Import Wizard, where both the movies and ratings tables were loaded for efficient querying and analysis.

Data Cleaning

The dataset was cleaned directly within MySQL Workbench to ensure data accuracy and consistency. The key steps included:

- Handling Missing Values – Removed or replaced records with NULL values in critical fields like movie title, genres, and ratings.

- Removing Duplicates – Eliminated duplicate entries based on unique identifiers such as movie_id.

- Standardizing Data – Ensured consistent formatting for titles, genres, and release years.

- Filtering Irrelevant Records – Excluded incomplete or irrelevant movies with missing duration or invalid ratings for better analysis.

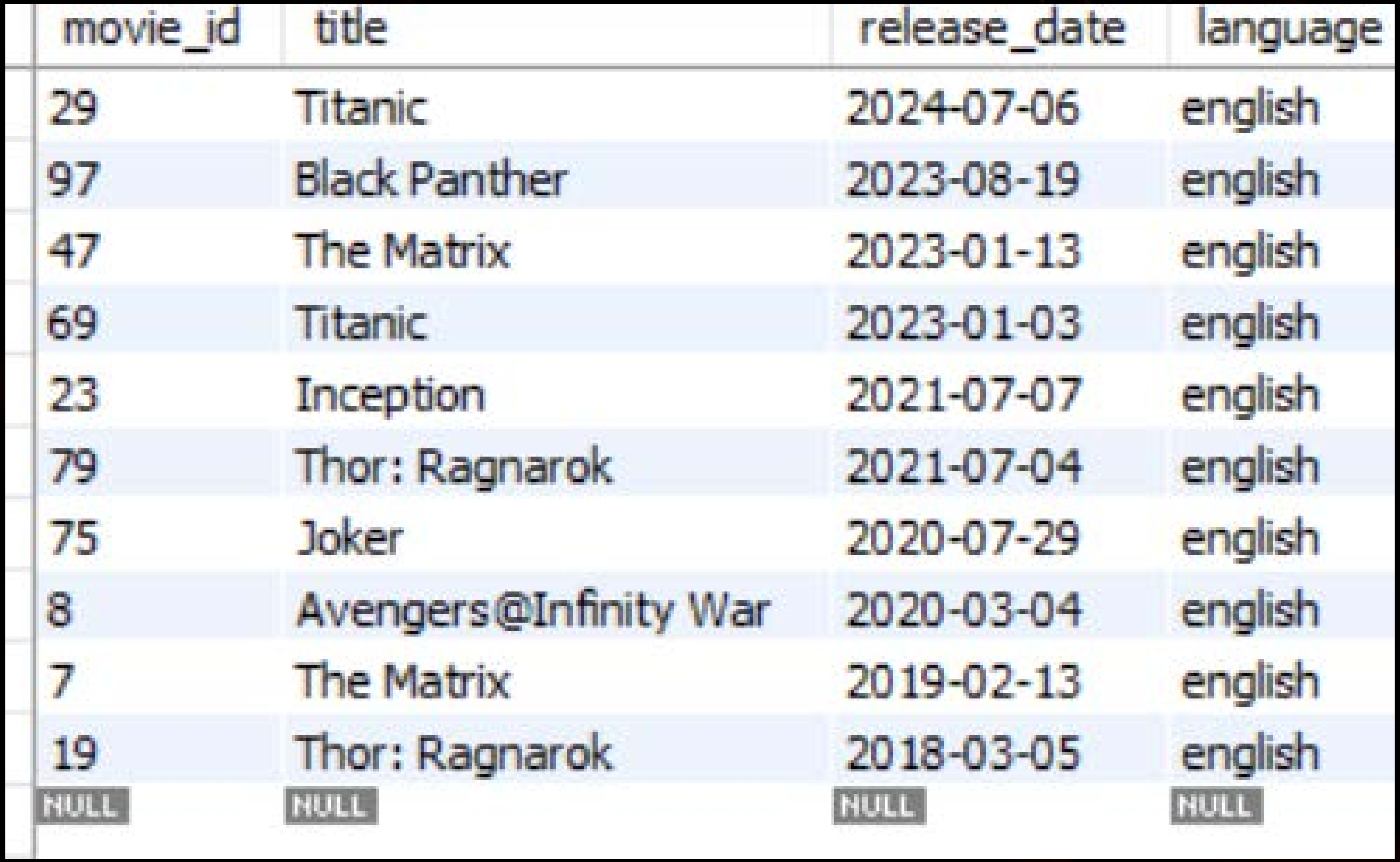
DATA ANALYSIS

Problem Statements

01

Show the 10 most recently released English movies.

```
SELECT movie_id, title, release_date, language
FROM movies
ORDER BY release_date DESC
LIMIT 10;
```



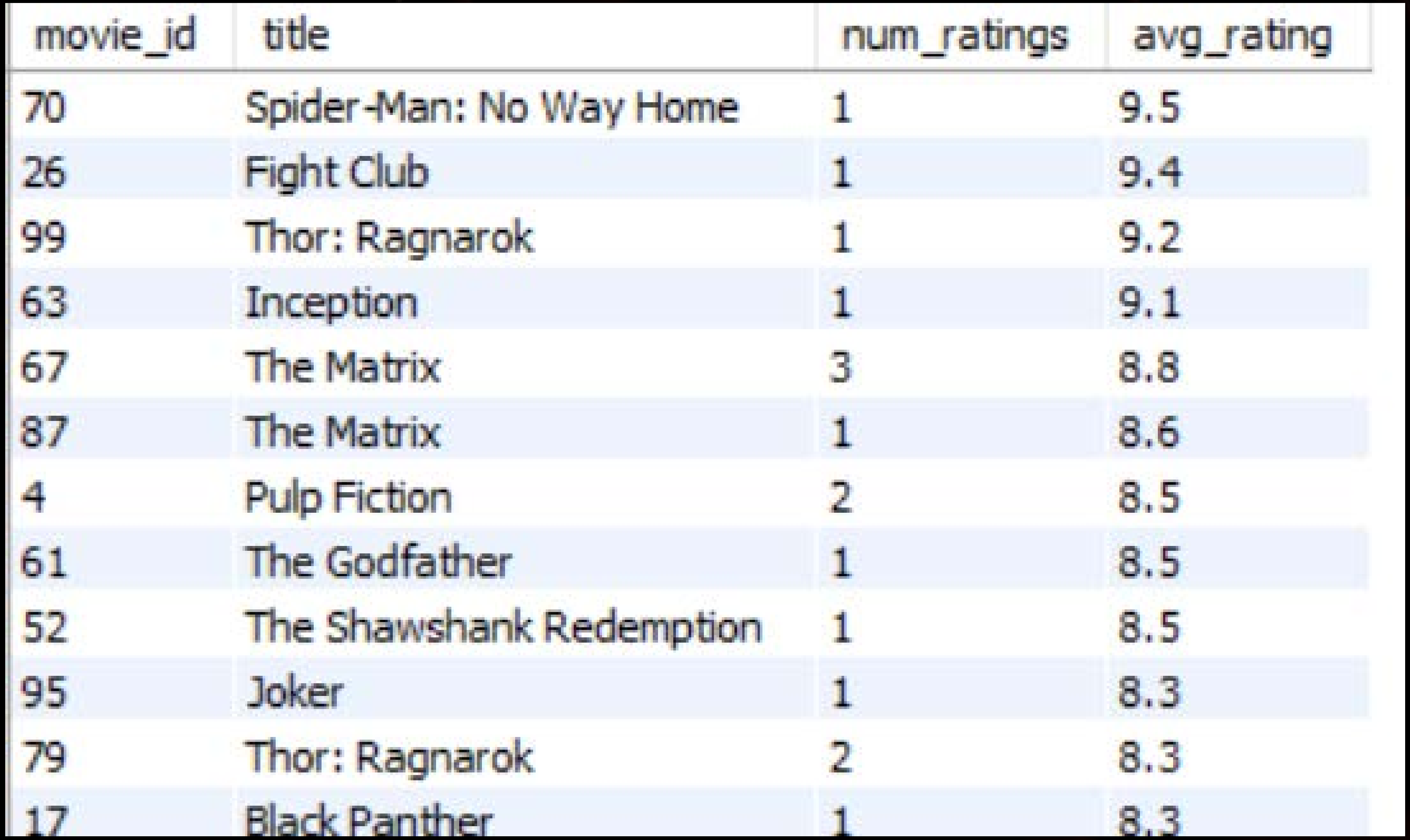
movie_id	title	release_date	language
29	Titanic	2024-07-06	english
97	Black Panther	2023-08-19	english
47	The Matrix	2023-01-13	english
69	Titanic	2023-01-03	english
23	Inception	2021-07-07	english
79	Thor: Ragnarok	2021-07-04	english
75	Joker	2020-07-29	english
8	Avengers@Infinity War	2020-03-04	english
7	The Matrix	2019-02-13	english
19	Thor: Ragnarok	2018-03-05	english
NULL	NULL	NULL	NULL

Problem Statements

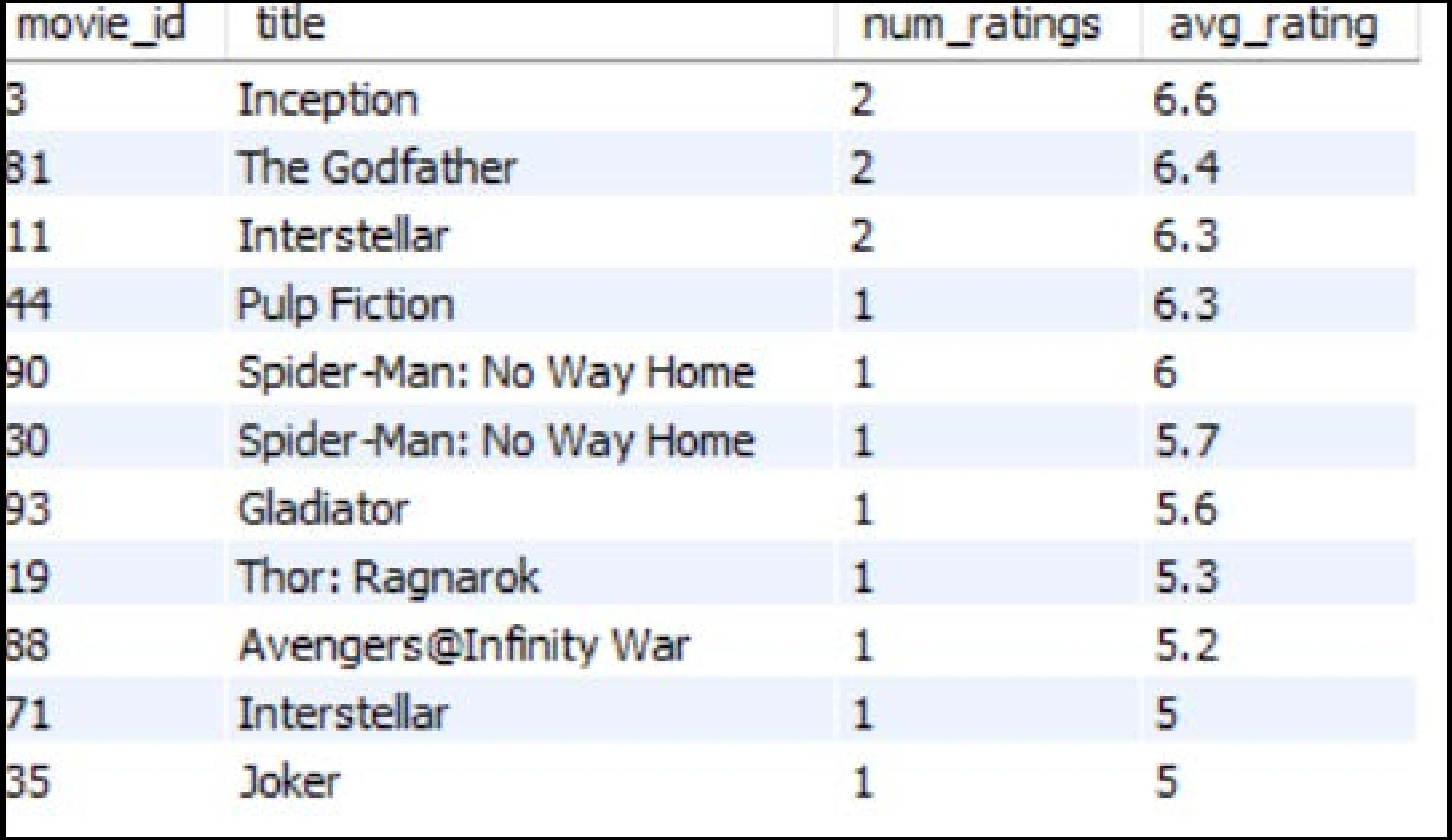
02

For each movie, get title, number of ratings, and average rating.

```
SELECT m.movie_id, m.title,
 count(r.rating) AS num_ratings,
   ROUND(AVG(r.rating), 1) AS avg_rating
FROM movies AS m
JOIN ratings AS r ON m.movie_id=r.movie_id
GROUP BY m.movie_id, m.title
ORDER BY avg_rating DESC;
```



movie_id	title	num_ratings	avg_rating
70	Spider-Man: No Way Home	1	9.5
26	Fight Club	1	9.4
99	Thor: Ragnarok	1	9.2
63	Inception	1	9.1
67	The Matrix	3	8.8
87	The Matrix	1	8.6
4	Pulp Fiction	2	8.5
61	The Godfather	1	8.5
52	The Shawshank Redemption	1	8.5
95	Joker	1	8.3
79	Thor: Ragnarok	2	8.3
17	Black Panther	1	8.3



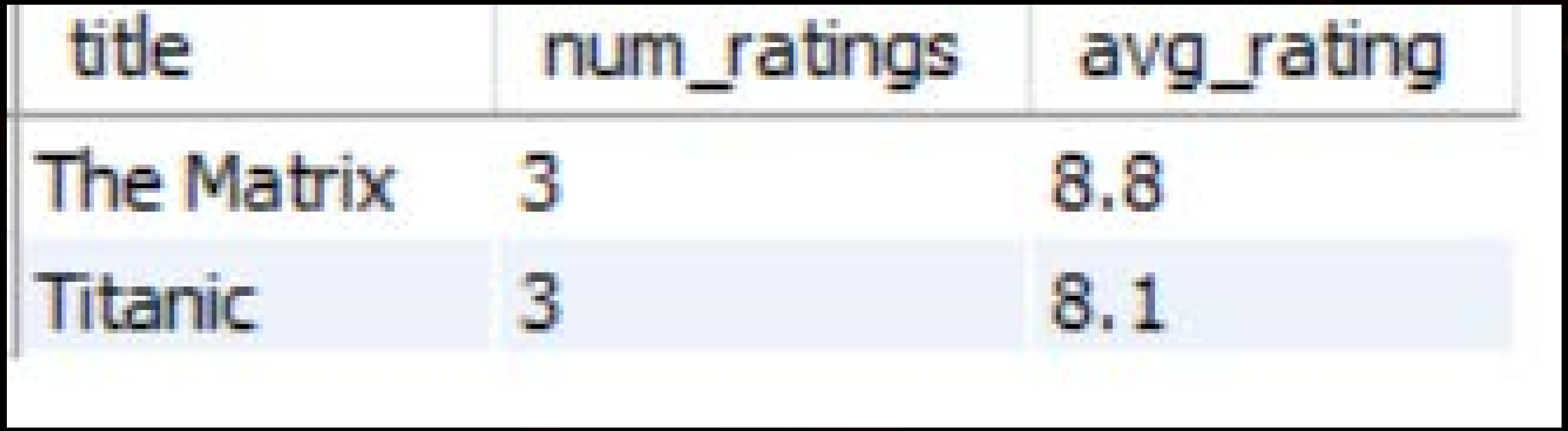
movie_id	title	num_ratings	avg_rating
3	Inception	2	6.6
31	The Godfather	2	6.4
11	Interstellar	2	6.3
44	Pulp Fiction	1	6.3
90	Spider-Man: No Way Home	1	6
30	Spider-Man: No Way Home	1	5.7
93	Gladiator	1	5.6
19	Thor: Ragnarok	1	5.3
38	Avengers@Infinity War	1	5.2
71	Interstellar	1	5
35	Joker	1	5

Problem Statements

03

Movies with strong evidence: avg ≥ 4.5 and at least 3 ratings.

```
SELECT m.title, COUNT(r.rating) AS
num_ratings, ROUND(AVG(r.rating),1) AS
avg_rating
FROM movies AS m
JOIN ratings AS r ON m.movie_id=r.movie_id
GROUP BY m.movie_id, m.title
HAVING num_ratings >= 3 AND avg_rating >=
4.5
ORDER BY avg_rating DESC;
```

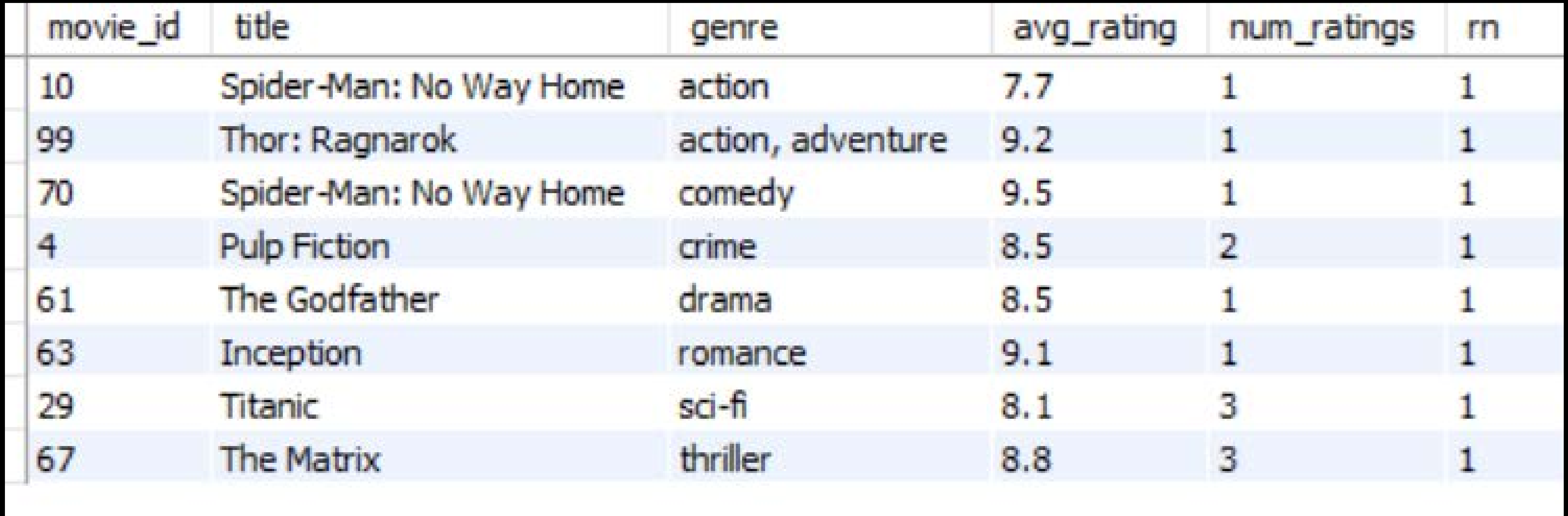


title	num_ratings	avg_rating
The Matrix	3	8.8
Titanic	3	8.1

Problem Statements

04 Top movie per genre.

```
WITH movie_stats AS (
  SELECT m.movie_id, m.title, m.genre,
         ROUND(AVG(r.rating), 1) AS avg_rating,
         COUNT(r.rating) AS num_ratings
  FROM movies AS m
  LEFT JOIN ratings AS r ON m.movie_id = r.movie_id
  GROUP BY m.movie_id, m.title, m.genre
)
SELECT movie_id, title, genre, avg_rating, num_ratings, rn
FROM (
  SELECT *,
         ROW_NUMBER() OVER (PARTITION BY genre ORDER BY avg_rating DESC, num_ratings DESC) AS rn
  FROM movie_stats
) AS t
WHERE rn = 1;
```



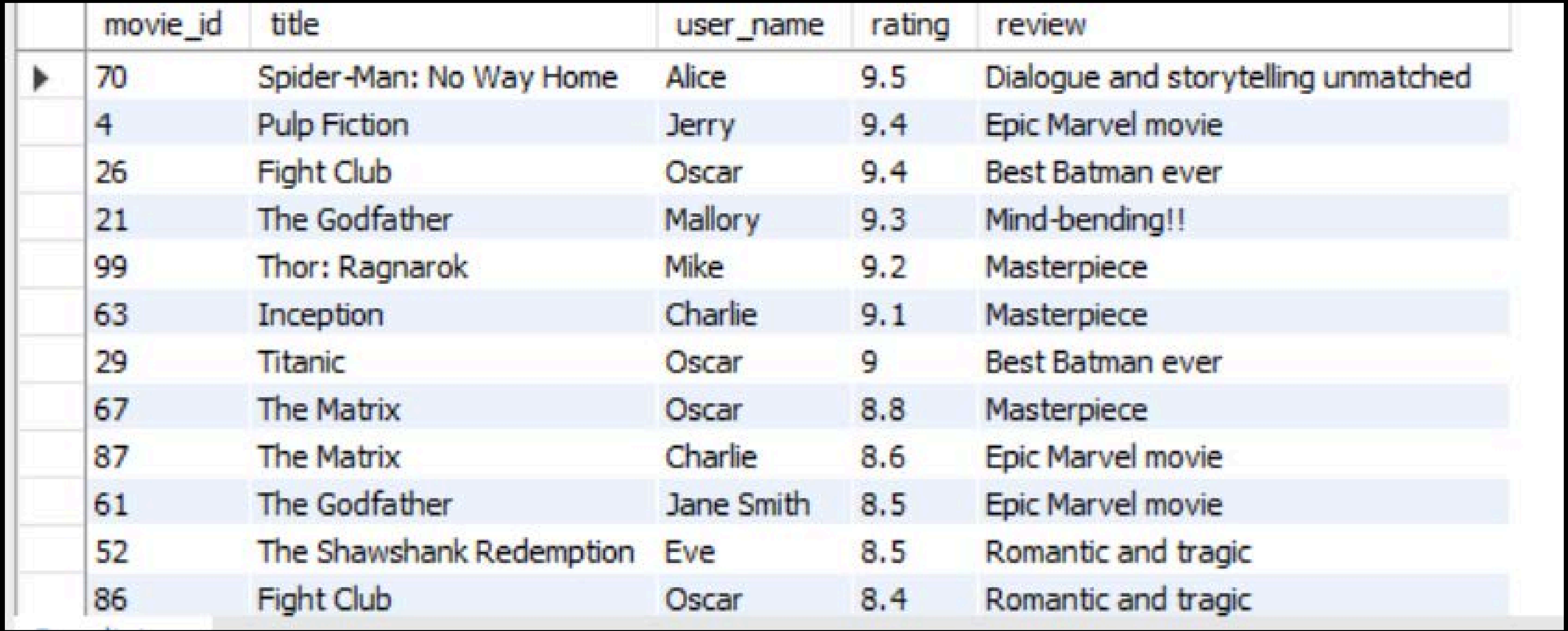
movie_id	title	genre	avg_rating	num_ratings	rn
10	Spider-Man: No Way Home	action	7.7	1	1
99	Thor: Ragnarok	action, adventure	9.2	1	1
70	Spider-Man: No Way Home	comedy	9.5	1	1
4	Pulp Fiction	crime	8.5	2	1
61	The Godfather	drama	8.5	1	1
63	Inception	romance	9.1	1	1
29	Titanic	sci-fi	8.1	3	1
67	The Matrix	thriller	8.8	3	1

Problem Statements

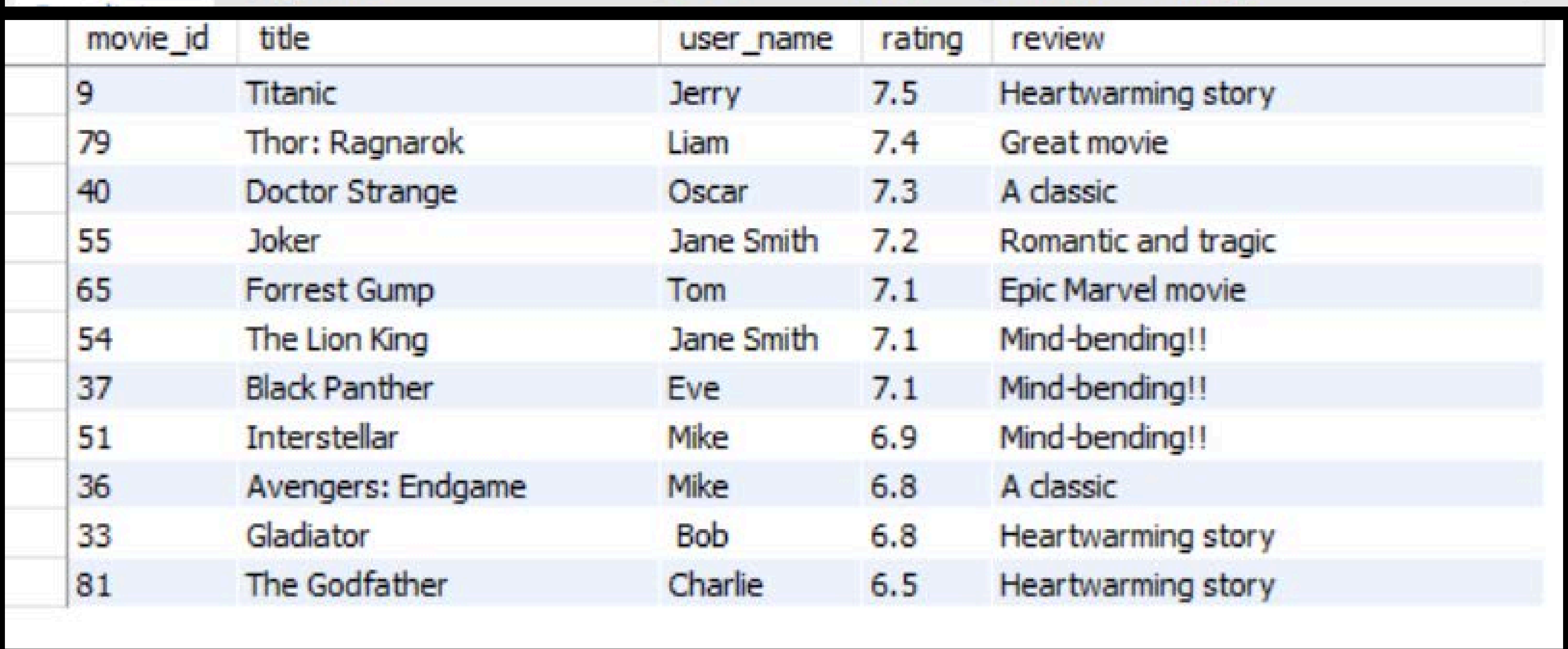
05

Show the 30 latest review per movie.

```
WITH ranked AS (
  SELECT r.*,
         ROW_NUMBER() OVER (PARTITION BY
movie_id ORDER BY rating_id DESC) AS rn
  FROM ratings AS r
)
SELECT m.movie_id, m.title, ranked.user_name,
ranked.rating, ranked.review
FROM movies AS m
JOIN ranked ON m.movie_id = ranked.movie_id
WHERE ranked.rn = 1
ORDER BY rating DESC
LIMIT 30;
```



movie_id	title	user_name	rating	review
70	Spider-Man: No Way Home	Alice	9.5	Dialogue and storytelling unmatched
4	Pulp Fiction	Jerry	9.4	Epic Marvel movie
26	Fight Club	Oscar	9.4	Best Batman ever
21	The Godfather	Mallory	9.3	Mind-bending!!
99	Thor: Ragnarok	Mike	9.2	Masterpiece
63	Inception	Charlie	9.1	Masterpiece
29	Titanic	Oscar	9	Best Batman ever
67	The Matrix	Oscar	8.8	Masterpiece
87	The Matrix	Charlie	8.6	Epic Marvel movie
61	The Godfather	Jane Smith	8.5	Epic Marvel movie
52	The Shawshank Redemption	Eve	8.5	Romantic and tragic
86	Fight Club	Oscar	8.4	Romantic and tragic



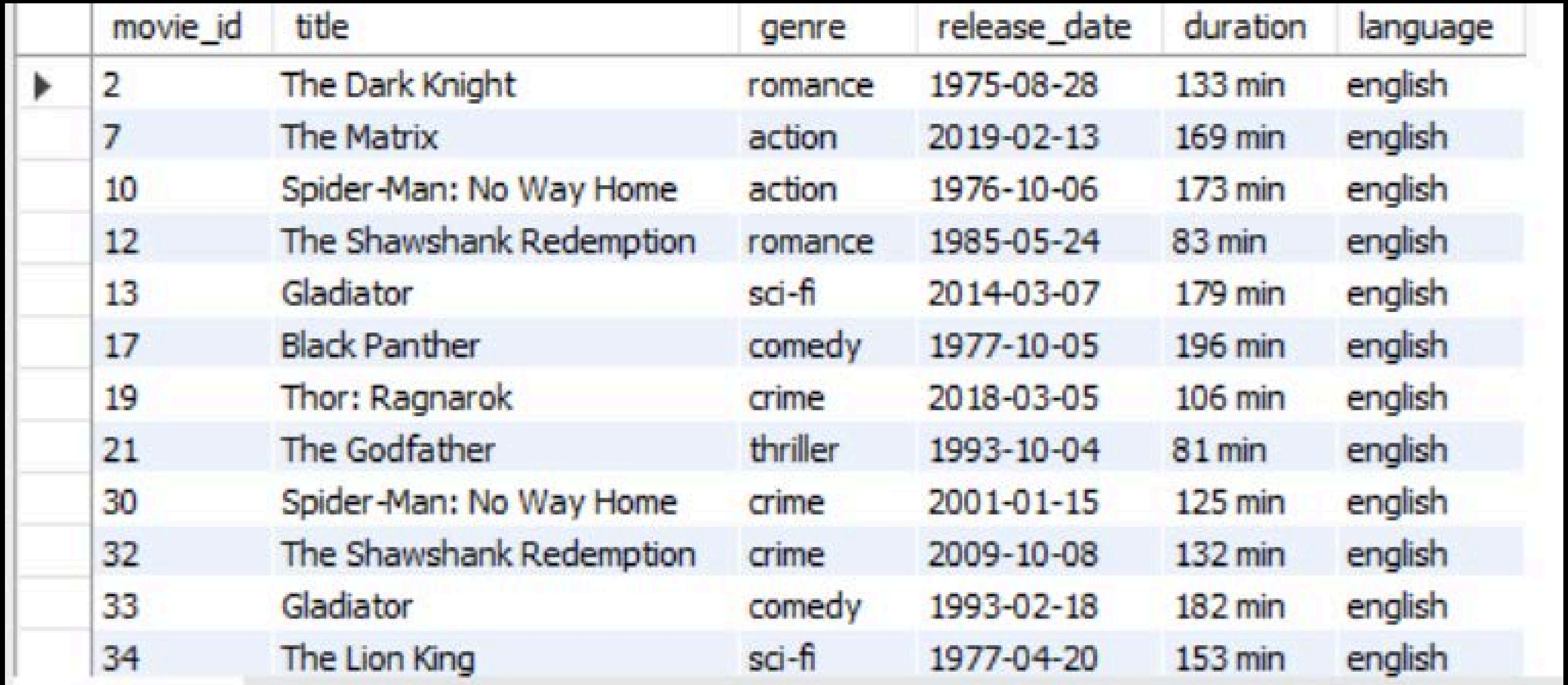
movie_id	title	user_name	rating	review
9	Titanic	Jerry	7.5	Heartwarming story
79	Thor: Ragnarok	Liam	7.4	Great movie
40	Doctor Strange	Oscar	7.3	A classic
55	Joker	Jane Smith	7.2	Romantic and tragic
65	Forrest Gump	Tom	7.1	Epic Marvel movie
54	The Lion King	Jane Smith	7.1	Mind-bending!!
37	Black Panther	Eve	7.1	Mind-bending!!
51	Interstellar	Mike	6.9	Mind-bending!!
36	Avengers: Endgame	Mike	6.8	A classic
33	Gladiator	Bob	6.8	Heartwarming story
81	The Godfather	Charlie	6.5	Heartwarming story

Problem Statements

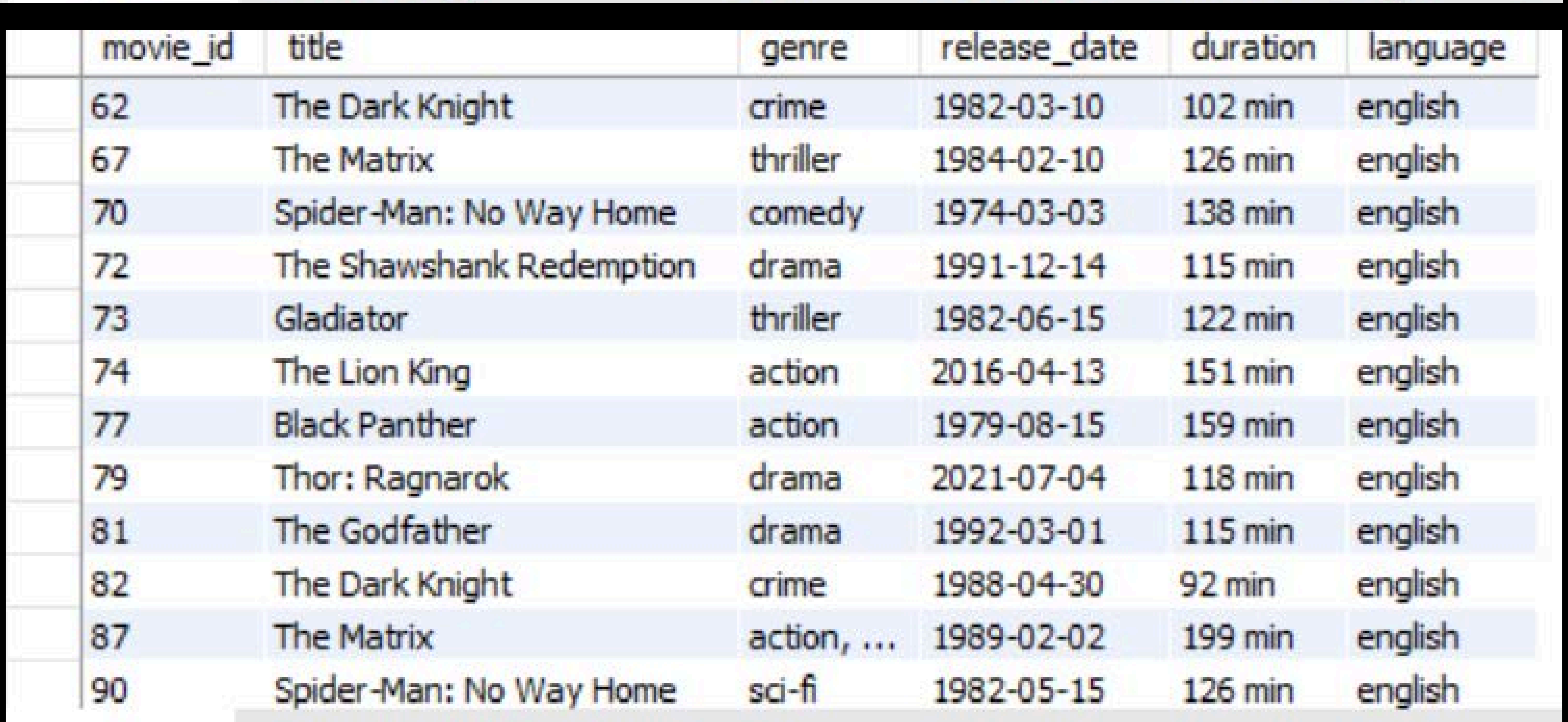
06

Find all movies whose titles contain "Matrix", start with "The", end with "Redemption", or contain the letter "a" more than once.

```
SELECT *
FROM movies
WHERE title LIKE '%Matrix%'
   OR title LIKE 'The%'
   OR title LIKE '%Redemption'
   OR LENGTH(title) -
LENGTH(REPLACE(title, 'a', '')) > 1
```



movie_id	title	genre	release_date	duration	language
2	The Dark Knight	romance	1975-08-28	133 min	english
7	The Matrix	action	2019-02-13	169 min	english
10	Spider-Man: No Way Home	action	1976-10-06	173 min	english
12	The Shawshank Redemption	romance	1985-05-24	83 min	english
13	Gladiator	sci-fi	2014-03-07	179 min	english
17	Black Panther	comedy	1977-10-05	196 min	english
19	Thor: Ragnarok	crime	2018-03-05	106 min	english
21	The Godfather	thriller	1993-10-04	81 min	english
30	Spider-Man: No Way Home	crime	2001-01-15	125 min	english
32	The Shawshank Redemption	crime	2009-10-08	132 min	english
33	Gladiator	comedy	1993-02-18	182 min	english
34	The Lion King	sci-fi	1977-04-20	153 min	english



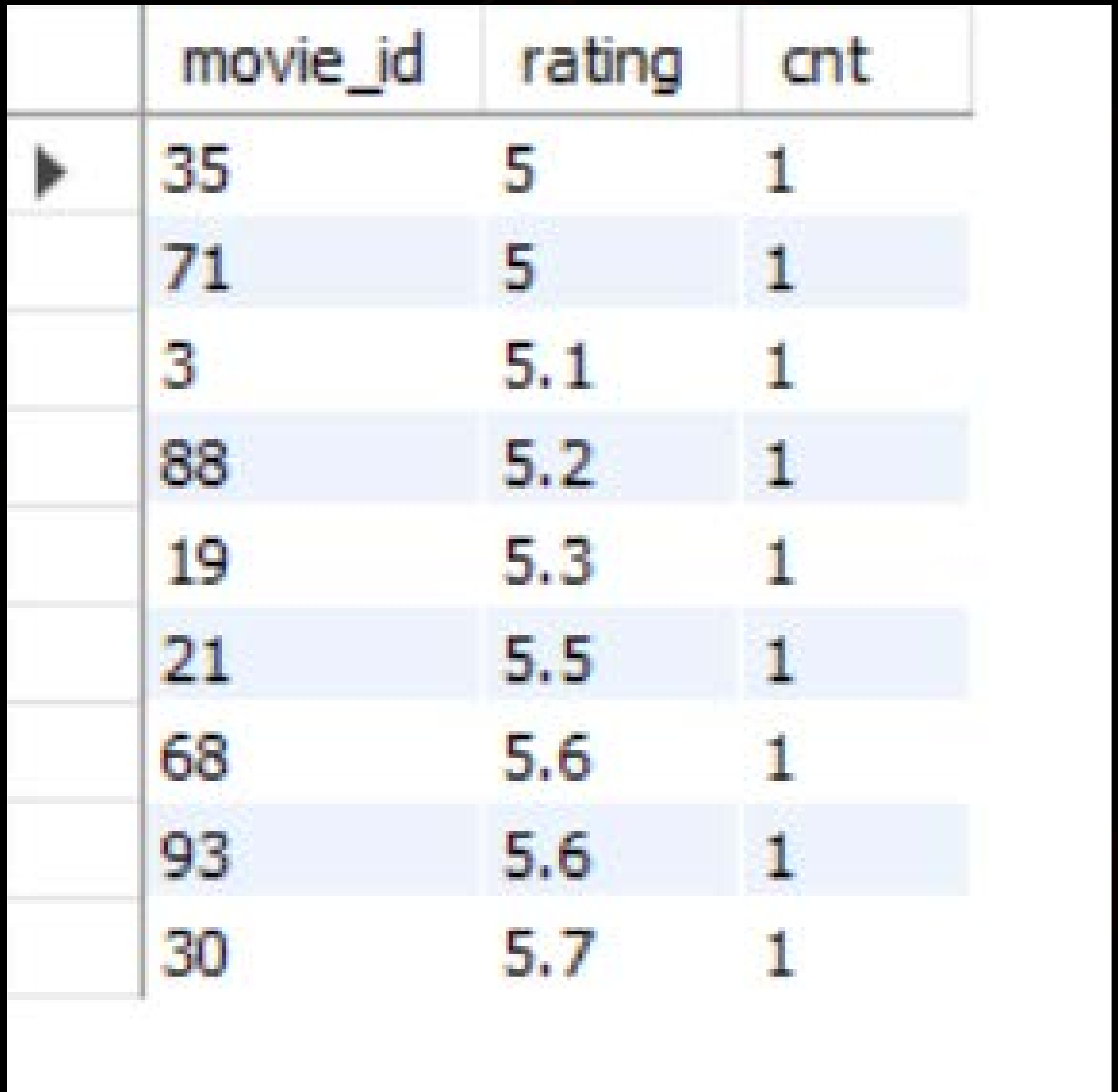
movie_id	title	genre	release_date	duration	language
62	The Dark Knight	crime	1982-03-10	102 min	english
67	The Matrix	thriller	1984-02-10	126 min	english
70	Spider-Man: No Way Home	comedy	1974-03-03	138 min	english
72	The Shawshank Redemption	drama	1991-12-14	115 min	english
73	Gladiator	thriller	1982-06-15	122 min	english
74	The Lion King	action	2016-04-13	151 min	english
77	Black Panther	action	1979-08-15	159 min	english
79	Thor: Ragnarok	drama	2021-07-04	118 min	english
81	The Godfather	drama	1992-03-01	115 min	english
82	The Dark Knight	crime	1988-04-30	92 min	english
87	The Matrix	action, ...	1989-02-02	199 min	english
90	Spider-Man: No Way Home	sci-fi	1982-05-15	126 min	english

Problem Statements

07

Find the number of ratings for each rating value between 5 and 5.7 in the ratings table, and display the counts sorted by rating.

```
SELECT movie_id, rating, COUNT(*) AS cnt
FROM ratings
WHERE rating BETWEEN 5 AND 5.7
GROUP BY rating, movie_id
ORDER BY rating;
```



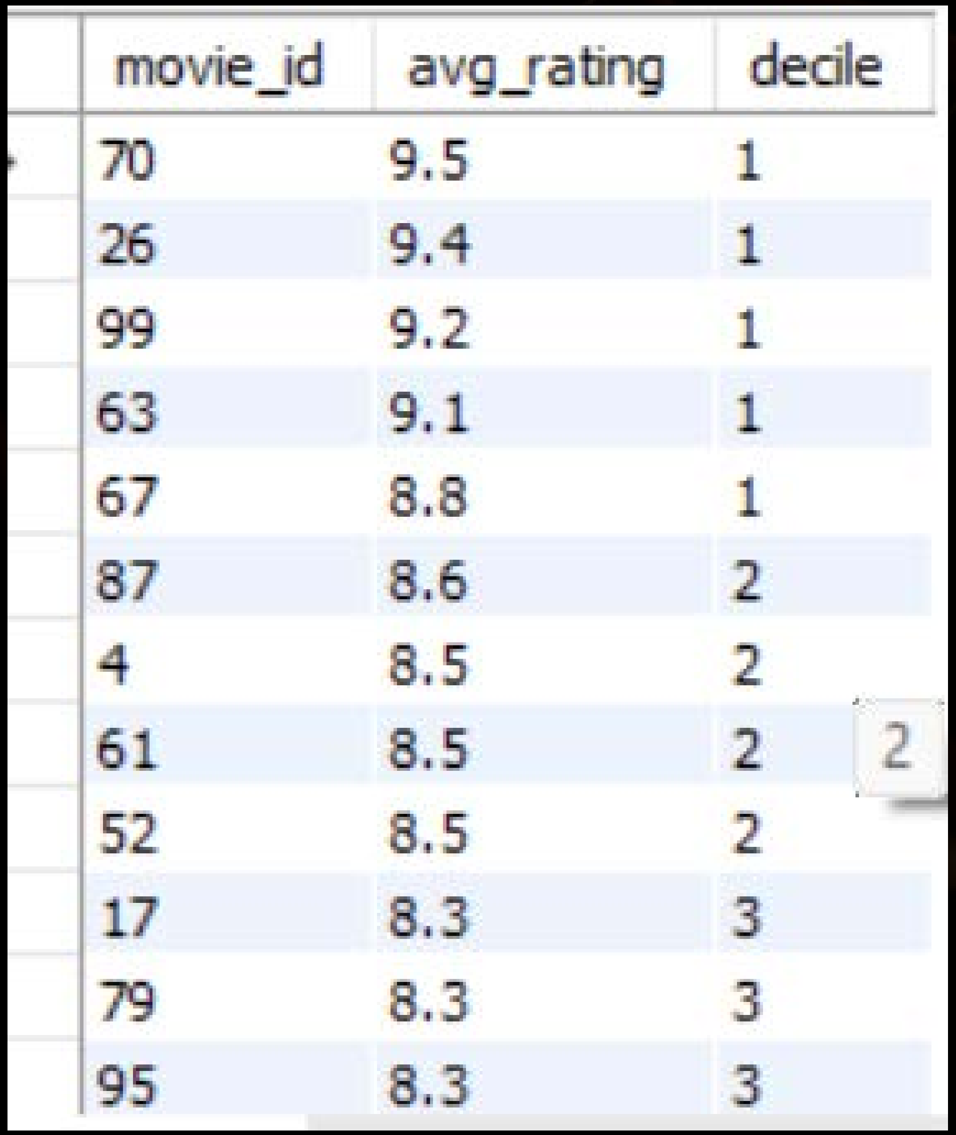
movie_id	rating	cnt
35	5	1
71	5	1
3	5.1	1
88	5.2	1
19	5.3	1
21	5.5	1
68	5.6	1
93	5.6	1
30	5.7	1

Problem Statements

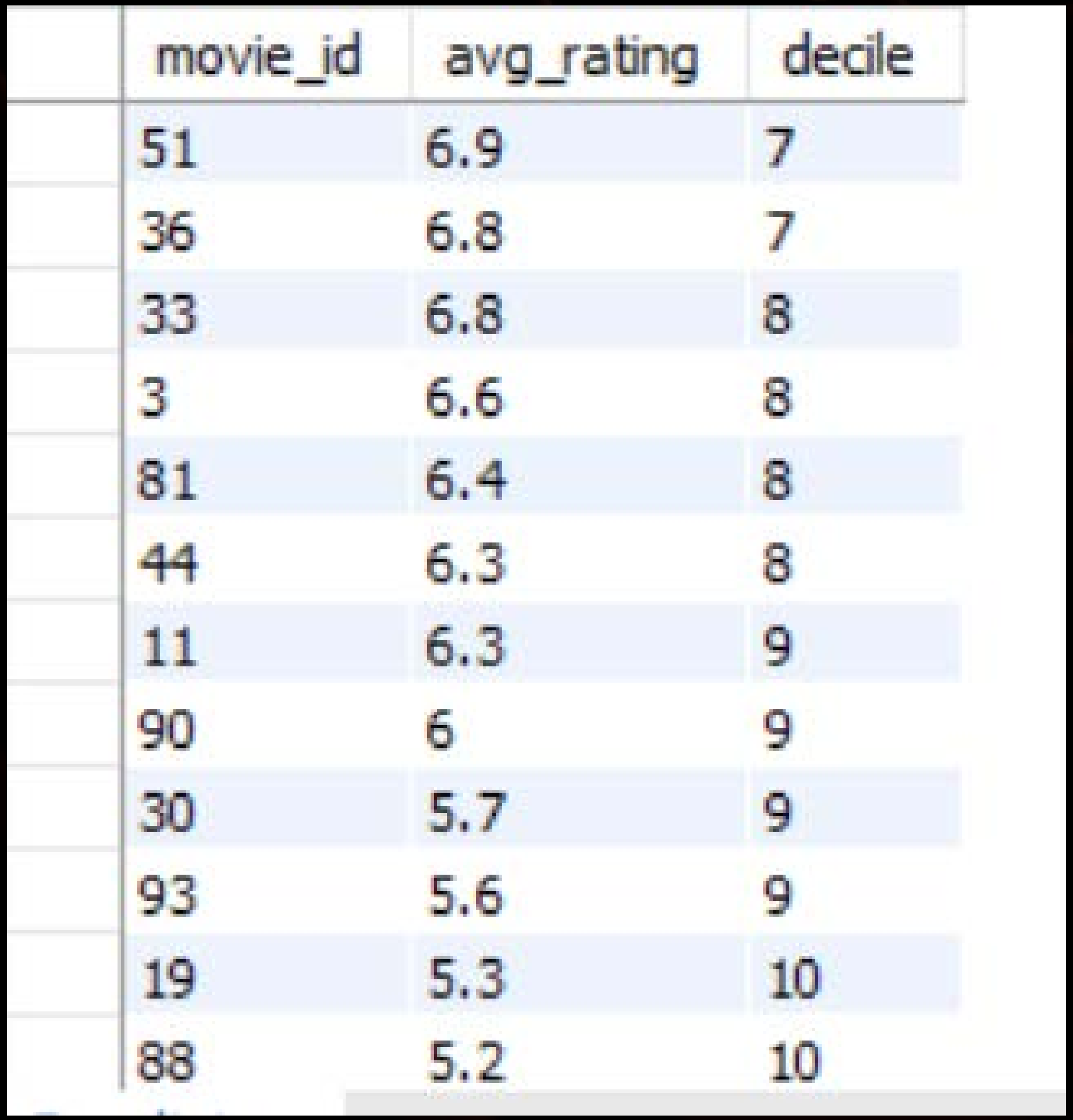
08

Assign each movie to a decile based on average rating (top 10% = decile 1).

```
 WITH movie_avg AS (
   SELECT
      movie_id,
      ROUND(AVG(rating), 1) AS avg_rating
   FROM ratings
   GROUP BY movie_id
)
SELECT
   movie_id,
   avg_rating,
   NTILE(10) OVER (ORDER BY avg_rating DESC)
AS decile
FROM movie_avg
ORDER BY decile, avg_rating DESC;
```



movie_id	avg_rating	decile
70	9.5	1
26	9.4	1
99	9.2	1
63	9.1	1
67	8.8	1
87	8.6	2
4	8.5	2
61	8.5	2
52	8.5	2
17	8.3	3
79	8.3	3
95	8.3	3



movie_id	avg_rating	decile
51	6.9	7
36	6.8	7
33	6.8	8
3	6.6	8
81	6.4	8
44	6.3	8
11	6.3	9
90	6	9
30	5.7	9
93	5.6	9
19	5.3	10
88	5.2	10

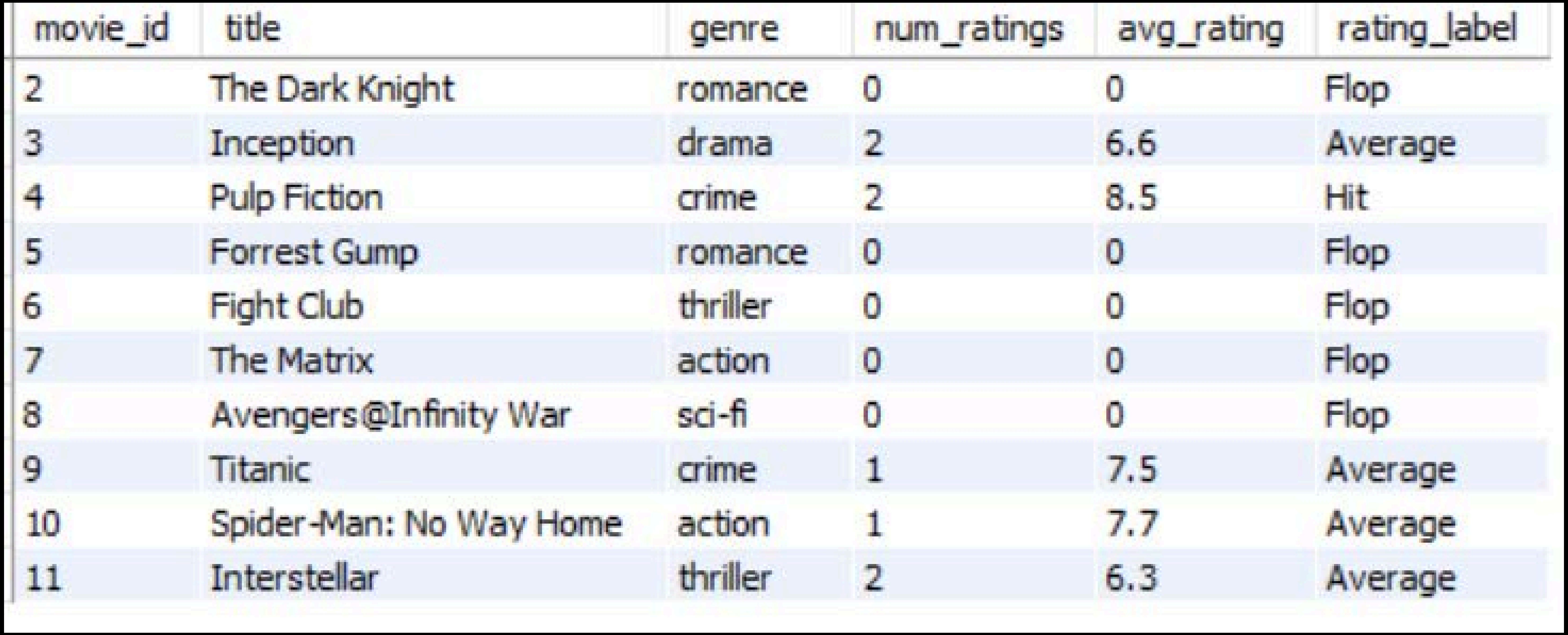
Problem Statements

09

Provide an easy-to-query summary view and classify movies as Hit/Average/Flop.

```
CREATE OR REPLACE VIEW
movie_rating_summary AS
SELECT
   m.movie_id,
   m.title,
   m.genre,
   COUNT(r.rating) AS num_ratings,
   ROUND(COALESCE(AVG(r.rating), 0), 1) AS
avg_rating,
   CASE
      WHEN COALESCE(AVG(r.rating), 0) >= 8.0
THEN 'Hit'
      WHEN COALESCE(AVG(r.rating), 0) >= 5.7
THEN 'Average'
      ELSE 'Flop'
   END AS rating_label
FROM movies m
LEFT JOIN ratings r ON m.movie_id = r.movie_id
GROUP BY m.movie_id, m.title, m.genre;

-- To see the top 10 records
SELECT * FROM movie_rating_summary
LIMIT 10;
```

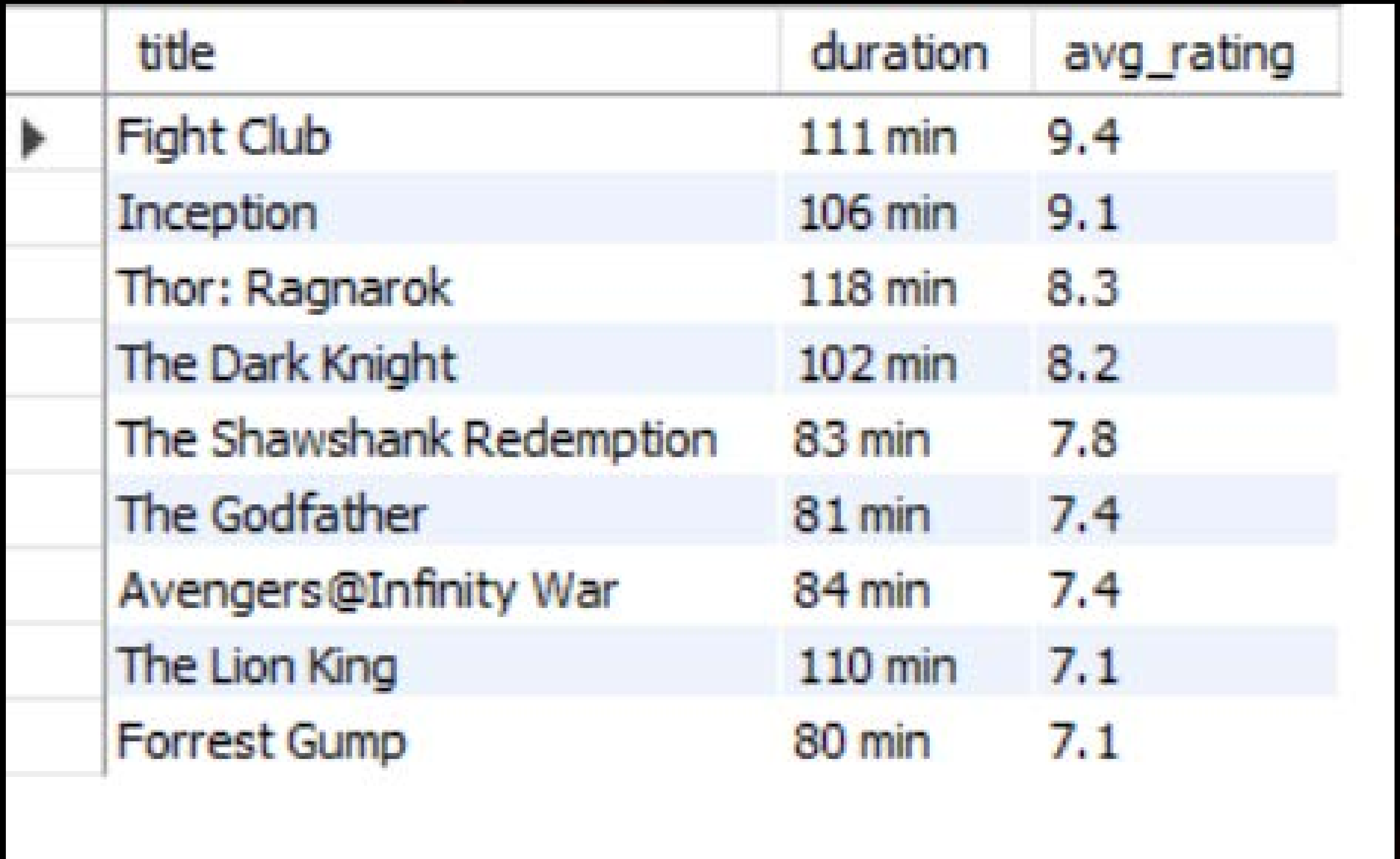


movie_id	title	genre	num_ratings	avg_rating	rating_label
2	The Dark Knight	romance	0	0	Flop
3	Inception	drama	2	6.6	Average
4	Pulp Fiction	crime	2	8.5	Hit
5	Forrest Gump	romance	0	0	Flop
6	Fight Club	thriller	0	0	Flop
7	The Matrix	action	0	0	Flop
8	Avengers@Infinity War	sci-fi	0	0	Flop
9	Titanic	crime	1	7.5	Average
10	Spider-Man: No Way Home	action	1	7.7	Average
11	Interstellar	thriller	2	6.3	Average

Problem Statements

10

Problem: Show movies shorter than 120 mins with average rating ≥ 7.



	title	duration	avg_rating
▶	Fight Club	111 min	9.4
	Inception	106 min	9.1
	Thor: Ragnarok	118 min	8.3
	The Dark Knight	102 min	8.2
	The Shawshank Redemption	83 min	7.8
	The Godfather	81 min	7.4
	Avengers@Infinity War	84 min	7.4
	The Lion King	110 min	7.1
	Forrest Gump	80 min	7.1

```
SELECT m.title, m.duration,
ROUND(AVG(r.rating),1) AS avg_rating
FROM movies m
JOIN ratings r ON m.movie_id = r.movie_id
WHERE m.duration < 120
GROUP BY m.movie_id, m.title
HAVING avg_rating >= 7
ORDER BY avg_rating DESC;
```

Insights & Findings

- Identified the 10 most recent English movies along with their ratings, top-performing genres, and highly rated short movies (<120 mins).
- Highlighted strong performers with avg ≥ 4.5 and ≥ 3 ratings, and classified movies into Hit, Average, and Flop categories.
- Analyzed latest reviews, rating distributions, and title-based trends to understand audience preferences effectively.

Business Impact

- Enhanced Content Relevance

Identifying the 10 most recently released English movies with high ratings helps target trending and quality content, improving audience engagement.

- Personalized Recommendations

Leveraging insights like top movies per genre and classifying movies as Hit/Average/Flop enables better personalization for viewers, boosting satisfaction.

- Operational Efficiency

Filtering movies based on duration (<120 mins) and strong ratings ensures optimized content selection, leading to improved viewer retention.

- Informed Decision-Making

Analyzing ratings distribution, decile-based classifications, and reviews supports data-driven strategies for content acquisition, marketing, and resource allocation.

Thank You

For exploring this presentation on the IMDB Data Analysis Project.
Your interest and engagement are greatly appreciated as we continue to uncover valuable insights from the data and support data-driven decision-making in the entertainment industry.