

GRAPH AND MODERN DATABASES

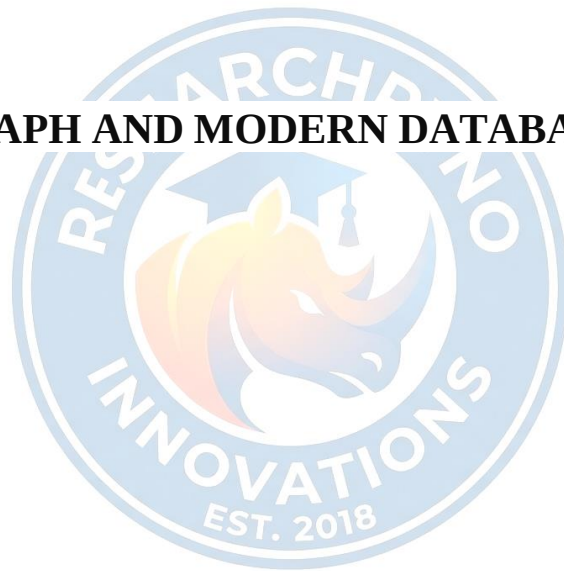


Table of Contents

Part 1: Peer Review of the Research Paper	6
1. Summary	6
2. Major Issues	6
3. Minor Issues	7
Part 2. Implementation	9
Assignment 1	9
Implementation	9
Rationale of design decisions	11
Benefits and drawbacks	11
Assignment 2	12
Implimentation	12
The rationale of design decisions	17
Benefits and drawbacks	17
Assignment 3.....	18
Implementation	18
Rationale of design decisions	22
Benefits and drawbacks	23
Assignment 4	23
Implementation.....	23
The rationale of design decisions	32
Benefits and drawbacks	32
Assignment 5.....	33
Implementation.....	33
The rationale of design decisions	42

Benefits and drawbacks	42
References	43
Appendix	45



Table of Figures

Figure 1: Add directors, ratings, movies, genres.....	9
Figure 2: Query 1	9
Figure 3: Query 2, Get the rating of 'Pulp Fiction'	10
Figure 4: Query 3 Get the genre of 'Jurassic Park'	10
Figure 5: Query 4, Set a new rating for 'Inception'	10
Figure 6: Query 5, Increment the rating of 'Pulp Fiction' by 0.5.....	10
Figure 7: Query 6: Get all movie names	10
Figure 8: Query 7: Check if 'The Dark Knight' exists in the database	11
Figure 10: Query 8: Delete the genre of 'Inception'	11
Figure 11: Query 9: Get the director and genre of 'Pulp Fiction'	11
Figure 12: Query 10: Get the average rating of all movies	11
Figure 13: Create Database	12
Figure 14: Create a Table of Movies.....	12
Figure 15: Insert Value of movies.....	13
Figure 16: The Select Query	13
Figure 17: Retrieve the average rating of all movies	14
Figure 18: Retrieve movie details by title	14
Figure 19: Retrieve movies released in a specific year	14
Figure 20: Retrieve movies directed by a specific director	14
Figure 21: Retrieve movies with a rating higher than a certain value	15
Figure 21: Retrieve movies within a specific genre	15
Figure 22: Retrieve movies with a specific runtime or shorter	16
Figure 23: Retrieve movies released between two years	16
Figure 24: Retrieve movies directed by Christopher Nolan released after the year 2000	17
Figure 25: Movie database JSON schema.....	18
Figure 26: Movie database JSON.....	19
Figure 27: Python code for checking JSON schema validation	20
Figure 28: Output of validation of JSON schema	22
Figure 29: Create a database	23
Figure 30. Create four table.....	23

Figure 31 : Insert movie name.....	24
Figure 32: Insert director name	24
Figure 33:Insert ratings	25
Figure 34:Insert genres.....	25
Figure 35: Find all movies directed by Steven Spielberg.....	26
Figure 36: Find the average rating of movies released after 2000	26
Figure 37: Find all movies with the "Action" genre.....	27
Figure 38: Find the directors who have won more than one Academy Award	27
Figure 39: Find the top 3 highest rated movies	28
Figure 40: Find the total number of voters for each genre.....	30
Figure 41: Find the movie with the highest number of voters	31
Figure 26 :Find the average budget of movies directed by American directors.....	31
Figure 42: Find the total budget spent on movies in the "Crime" genre	31
Figure 43: Find the average rating of movies directed by British directors	32
Figure 44: Creating Movie table and inserting value in this	33
Figure 45: Graphical show for movie table data	33
Figure 46: Creating Director table and inserting value in this	34
Figure 47: Graphical show for director table data.....	34
Figure 48: Creating Rating table and inserting value in this	35
Figure 49: Graphical show for rating table data.....	35
Figure 50: Connecting tables.....	36
Figure 51: Show the entire graph	36
Figure 52: Show Movies and their Ratings.....	37
Figure 53: Show the graph around a specific Movie (e.g., 'Inception').....	38
Figure 54: Showing movies released in 1994.....	38
Figure 55: Find movies with a specific genre: Action	39
Figure 56: Movies with the highest ratings for each Genre	40
Figure 57: Show the average rating for each Genre	41
Figure 58: Show the Distribution of Ratings for 'The Shawshank Redemption'	41
Figure 59: Showing crime movies	41
Figure 60: Show 5 low-rated movies in the movie table.....	42

Part 1: Peer Review of the Research Paper

1. Summary

Positive Feedback

A well-mannered and precise description of the NoSQL database has been described in the journal named *“Comparison between NoSQL and SQL databases”*. The history of NoSQL, the different types of NoSQL and the differences between NoSQL and SQL are also described in a comprehensive manner in the journal paper (Debrouvieret *al.*, 2021). The advantages and disadvantages of NoSQL have been described also in the journal with different real-world examples and the application of NoSQL is also mentioned with proper examples.

Findings of paper

The evolution and significance of the NoSQL database have been described in the journal titled *“Comparison between NoSQL and SQL databases”*. Different steps for the evolution of the NoSQL database have been described in this journal. NoSQL database has been categorized into four types and the data modelling approaches are also described also in the journal paper.

Existing Literature

The importance of growing interest and the volume of the database have been described in the existing literature in the journal named *“Comparison between NoSQL and SQL databases”*. Real-world applications of the NoSQL database have been described and limitations of relational databases are also mentioned in the journal. The scalability of performance and benefits of using a NoSQL database are described with social media marketing of the NoSQL database in the existing literature.

Significance of work

This article is significant on the grounds that it gives an intensive review of NoSQL databases, including data about their sorts, chronicles, and uses. It reveals insight into how NoSQL databases are changing with regard to contemporary information on the board by offering detailed information on their advantages and downsides in contrast with social databases (Bestaet *al.*, 2023). The review's creativity comes from its intensive assessment of certifiable circumstances, which feature its viable relevance. One such model is the joining of NoSQL in social communication platforms.

2. Major Issues

Major flaws

The paper's inability to fundamentally break down the limitations of NoSQL databases is one of its primary shortcomings. It would be useful to have an all more fair conversation on significant information base administration concerns like consistency, exchange backing, and information trustworthiness, even while it concedes a few limits such as confined language capacities and troubles with information examination. The paper's credibility is influenced by this omission, which could likewise bring about an absence of consciousness of the variation expected in choosing relational and NoSQL databases.

Similar work issue

The issue of a detailed review of the requirements of NoSQL databases is a huge drawback in the paper. It might profit from a more adjusted conversation on issues like consistency, exchange backing, and information trustworthiness, which are key variables in data set organization, even while it concedes specific constraints, like limited language capacities and hardships in information analysis. This omission degrades the paper's credibility and could leave peruses with an insufficient handle on the difficulties related to choosing social and NoSQL databases.

Challenges in current thinking

The constraints of NoSQL databases and their correlative capability related to social databases are featured by the creators in their discoveries, which give a test to conventional insight. They oppose despite the fact that NoSQL has benefits over social databases as far as speed and adaptability, it is less powerful in regions like dependability and information analysis. Moreover, there isn't a lot of information to back up this case — generally episodic proof from key parts on the lookout, including Instagram and Twitter.

Major Ethical Issues

Particularly in sensitive areas like virtual entertainment and online business, issues over the trustworthiness and constancy of information kept in NoSQL databases suggest moral contemplations. The review is profitable from a clearer analysis of the moral issues connected with the information the board methods, like the possible maltreatment of individual information and the meaning of liability and receptiveness in the treatment of client information. A more intensive handle of information base organization would come about because of resolving these moral issues and working on the paper's importance.

3. Minor Issues

Ambiguous meaning

In some places particularly while making sense of specialized thoughts and giving examples of NoSQL executions, the importance is confusing. This might be fixed by assuming that the researchers utilized less complex language, characterized terms all the more evidently, and gave more careful clarifications or guides to assist with making sense of troublesome ideas. Moreover, utilizing visual guides like flowcharts or outlines could work on the adequacy of enlightening significant thoughts.

Factual or numerical errors

The journal excludes exact quantitative information or numerical references for assessment, thusly there are no genuine, numerical, or unit defects that are apparent. Nonetheless, it's critical to ensure rightness and accuracy while introducing numerical or measurable information in scholarly composition. It is suggested that writers confirm numerical numbers, units of estimation, and factual calculations to forestall any errors that can misdirect readers or harm the paper's standing (Bonifatiet *al.*, 2020). Besides, a thorough reality-checking methodology and friend audit can help with spotting and fixing any potential errors before to distributing, ensuring the exactness and trustworthiness of the review results.

Lack of Citation

A little issue with the contextual investigation is that few citations — particularly those on the starting points and improvement of NoSQL databases — are not referred to. Despite the fact that the material could be right, giving due credit to the first sources will strengthen the paper's authenticity and scholarly thoroughness. Besides, a few segments could utilize more exact and compact expressing. Long sentences and complex phrasing could make specific passages challenging to examine and comprehend, particularly for perusers who are curious about the subject. Improving availability and commitment may be accomplished by working on language and isolating many-sided ideas into more modest, clearer pieces (Freitas *et al.*, 2020). Furthermore, the contextual analysis could profit from additional examination concerning current headways and creating designs in data set organization, providing perusers with a superior comprehension of how NoSQL innovations are created. Consolidating the present research results and industry upgrades would improve the talk and ensure the paper's relevance inside the structure of current data set approaches.

Part 2. Implementation

Assignment 1

Implementation

```
127.0.0.1:6379> SET director:1 "Christopher Nolan"  
OK  
127.0.0.1:6379> SET director:2 "Quentin Tarantino"  
OK  
127.0.0.1:6379> SET director:3 "Steven Spielberg"  
OK
```

```
127.0.0.1:6379> SET rating:2 8.8  
OK  
127.0.0.1:6379> SET rating:3 7.9  
OK  
127.0.0.1:6379> SET rating:1 9.5  
OK  
127.0.0.1:6379> SET movie:1 "Inception"  
OK  
127.0.0.1:6379> SET movie:2 "Pulp Fiction"  
OK  
127.0.0.1:6379> SET movie:3 "Jurassic Park"  
OK  
127.0.0.1:6379> SET genre:1 "Sci-Fi"  
OK  
127.0.0.1:6379> SET genre:2 "Crime"  
OK  
127.0.0.1:6379> SET genre:3 "Adventure"  
OK
```

Figure 1: Add directors, ratings, movies, genres

(Source: Redis)

The above image shows the Redis code for adding the names of the directors, the value for ratings, the name of the movies, and the genre of the movies.

```
127.0.0.1:6379> GET director:1  
"Christopher Nolan"  
127.0.0.1:6379>
```

Figure 2: Query 1

(Source: Redis)

Query 1 shows the code for getting the name of the director who made the movie Inception and here the name 'Christopher Nolan' is also shown.

```
127.0.0.1:6379> GET rating:2  
"8.8"
```

Figure 3: Query 2, Get the rating of 'Pulp Fiction'

(Source: Redis)

Query 2 shows the code for getting the ratings for the movie Pulp Fiction and here the rating 8.8 is also shown.

```
127.0.0.1:6379> GET genre:3  
"Adventure"
```

Figure 4: Query 3 Get the genre of 'Jurassic Park'

(Source: Redis)

Query 3 shows the code for getting the genre for the movie Jurassic Park and here the name of the genre adventure is also shown.

```
127.0.0.1:6379> SET rating:1 9.8  
OK
```

Figure 5: Query 4, Set a new rating for 'Inception'

(Source: Redis)

Query 4 shows the code for setting a new rating for the movie inception and here the output 'ok' is also shown.

```
127.0.0.1:6379> INCRBYFLOAT rating:2 0.5  
"9.30000000000000071"
```

Figure 6: Query 5, Increment the rating of 'Pulp Fiction' by 0.5

(Source: Redis)

Query 5 shows the code for incrementing the ratings for the movie pulp fiction by 5 and here the output 9.30 is also shown.

```
127.0.0.1:6379> MGET movie:1 movie:2 movie:3  
1) "Inception"  
2) "Pulp Fiction"  
3) "Jurassic Park"
```

Figure 7: Query 6: Get all movie names

(Source: Redis)

Query 6 shows the code for getting all the movie names and here the output inception, pulp fiction, jurassic park are also shown.

```
127.0.0.1:6379> EXISTS movie:4  
(integer) 0
```

Figure 8: Query 7: Check if 'The Dark Knight' exists in the database

(Source: Redis)

Query 7 shows the code for checking if the movie name 'The Dark Knight' exists in the database or not and here the output shows 0 which means that there is no movie is named like that.

```
127.0.0.1:6379> DEL genre:1  
(integer) 1
```

Figure 10: Query 8: Delete the genre of 'Inception'

(Source: Redis)

Query 8 shows the code for deleting the genre that contains the movie name Inception and here the output shows 1 which means the genre is deleted.

```
127.0.0.1:6379> MGET director:2 genre:2  
1) "Quentin Tarantino"  
2) "Crime"
```

Figure 11: Query 9: Get the director and genre of 'Pulp Fiction'

(Source: Redis)

Query 9 shows the code for getting the name of the director name and the genre name of the movie Pulp Fiction and here the output shows the name of the director name and the genre.

```
127.0.0.1:6379> EVAL "return (tonumber(redis.call('GET', 'rating:1')) + tonumber(redis.call('GET', 'rating:2')) + tonumber(redis.call('GET', 'rating:3')))/3" 0  
(integer) 9  
127.0.0.1:6379>
```

Figure 12: Query 10: Get the average rating of all movies

(Source: Redis)

Query 9 shows the code for getting the average ratings of all movies and here shows the output value 9.

Rationale of design decisions

The movie database is included with the key identification entities and the identification entities are Movie name, directors, movie genre and rating of a movie (Sicariet *al.* 2022). Relationships between entities are maintained with the identification key in such a way that the queries can show proper determination among attributes.

Benefits and drawbacks

Different movie-related information are included in the database system and it is very easy to find movie-related information by the users from the database. The trends and patterns are also shown by the database in the film industry.

Maintaining accuracy is the main drawback in the movie database design. Frequently updating the system and accuracy maintenance is very difficult in this database system.

Assignment 2

Implimentation

```
Connected to Test Cluster at 127.0.0.1:9042.  
[cqlsh 5.0.1 | Cassandra 3.11.16 | CQL spec 3.4.4 | Native protocol v4]  
Use HELP for help.  
WARNING: pyreadline dependency missing. Install to enable tab completion.  
cqlsh> CREATE KEYSPACE IF NOT EXISTS movie_database  
... WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};  
cqlsh> USE movie_database;  
cqlsh:movie_database>  
cqlsh:movie_database>
```

Figure 13: Create Database

(Source: Cassandra)

Here, created a database named “*movie_database*” in the platform of Cassandra.

```
cqlsh:movie_database> CREATE TABLE movies (  
...     movie_id UUID PRIMARY KEY,  
...     title TEXT,  
...     release_year INT,  
...     genre TEXT,  
...     director TEXT,  
...     rating FLOAT,  
...     description TEXT,  
...     runtime INT  
... );
```

Figure 14: Create a Table of Movies

(Source: Cassandra)

Here, created a table name movie where the attributes are movie_id, title, release_year, genre, director, rating, description, and routine.

```

151 | The Departed
76309c5e-2510-4401-a1c5-2f197907e011 | The presidencies of Kennedy and Johnson, the events of Vietnam, Watergate and other historical events unfold from the perspective
of an Alabama man with an IQ of 75, whose only desire is to be reunited with his childhood sweetheart. | Robert Zemeckis | Drama | 8.8 | 1994 |
142 | Forrest Gump
b0c47f3a-d2de-4e2b-847e-7088b4100448 | men bond over a number of years, finding solace and eventual redemption through acts of common decency. | Frank Darabont | Drama | 9.3 | Two imprisoned
142 | The Shawshank Redemption
72b0b7f7-9204-49f6-806a-7f916d4df7219 | A meek Hobbit from the Shire and eight companions set out on a journey to destroy the powerful One Ring and save Middle-earth from the Dark Lord Sauron. | Peter Jackson | Adventure | 8.8 | 2003 |
178 | The Lord of the Rings: The Fellowship of the Ring
d16db38c-61ad-4ef3-9c02-304d3085cdc2 | When a beautiful stranger leads computer hacker Neo to a forbidding underworld, he discovers the shocking truth--the life he knows is the elaborate deception of an evil cyber-intelligence. | Lana Wachowski | Action | 8.7 | 1999 |
176 | The Matrix
03d6544b-c04c-4901-9e0e-87a71fd7c68 | detectives, a rookie and a veteran, hunt a serial killer who uses the seven deadly sins as his motives. | David Fincher | Crime | 8.5 | 1995 |
127 | Seven
60f6b85f-3662-4c02-b3a4-c068fa12bceb | A team of explorers travel through a wormhole in space in an attempt to ensure humanity's survival. | Christopher Nolan | Adventure | 8.6 | 2014 |
169 | Interstellar
75c66523-7055-41b9-a329-80414035ec44 | As an outcast, he seeks revenge against the corrupt emperor who murdered his family and sent him into slavery. | Ridley Scott | Action | 8.5 | 2000 |
155 | Gladiator
12c9801a-2a40-4402-898f-0891d1dddec02 | In German-occupied Poland during World War II, industrialist Oskar Schindler gradually becomes concerned for his Jewish workforce after witnessing their persecution by the Nazis. | Steven Spielberg | Biography | 8.9 | 1993 |
105 | Schindler's List
20e05c0f-5357-457e-9558-4b0ac445254a | The Dark Knight is a haunted film that leaps beyond its origins and becomes an engrossing tragedy. It creates characters we come to care about. | Christopher Nolan | Action | 9 | 2008 |
172 | The Dark Knight
06e07b5e-cd22-4ce1-9211-037ae01e536 | The early life and career of Vito Corleone in 1920s New York City is portrayed, while his son, Michael, expands and tightens his grip on the family crime syndicate. | Francis Ford Coppola | Crime | 9 | 1974 |
182 | The Godfather: Part II
e9b802cf-b34a-484f-ae7a-bb64e4311499 | a German industrialist who saved more than a thousand mostly Polish-Jewish refugees from the Holocaust by employing them in his factories during World War II. | Steven Spielberg | Biography | 8.9 | 1993 |
165 | Schindler's List
e8fcdcdcc-c7b4-4f01-bb67-0edc0f50f32c | men bond over a number of years, finding solace and eventual redemption through acts of common decency. | Frank Darabont | Drama | 9.3 | Two imprisoned
142 | The Shawshank Redemption
(30 rows)
cqlsh:movie_database>

```

Figure 15: Insert Value of movies
(Source: Cassandra)

Inserted the value of movies with 30 rows in the platform of Cassandra.

```

cqlsh:movie_database> select * from movies;

movie_id | description | director | genre | rating | release_year | runtime | title
-----+-----+-----+-----+-----+-----+-----+-----
e1fcfc9a-9ebe-4093-b0ab-88985b17b763 | The lives of guards on Death Row are affected by one of their charges: a black man accused of child murder and rape, yet who has a mysterious gift. | Frank Darabont | Crime | 8.6 | 1999 | 189 | The Green Mile
748220b4-7d20-4d4a-b0ea-16dfe8a0bf0b | Gandalf and Aragorn lead the World of Men against Sauron's army to draw his gaze from Frodo and Sam as they approach Mount Doom with the One Ring. | Peter Jackson | Adventure | 8.9 | 2003 | 201 | The Lord of the Rings: The Return of the King
3c10f301-36af-4a6f-9a10-787e9070814 | While Frodo and Sam edge closer to Mordor with the help of the shifty Gollum, the divided fellowship makes a stand against Sauron's new ally, Saruman, and his hordes of Isengard. | Peter Jackson | Adventure | 8.7 | 2002 | 129 | The Lord of the Rings: The Two Towers
13355cf4-3018-450e-b049-cb4f55-1747c | It is based on the 1986 novel of the same name by Winston Groom and stars Tom Hanks, Robin Wright, Gary Sinise, Mykelti Williamson and Sally Field. | Robert Zemeckis | Drama | 8.8 | 1994 | 142 | Forrest Gump

```

Figure 16: The Select Query
(Source: Cassandra)

Executed a query to convey all the details about movies from the table of 'movies'.

```

cqlsh:movie_database> SELECT AVG(rating) FROM movies;

system.avg(rating)
-----
8.76

(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:movie_database>

```

Figure 17: Retrieve the average rating of all movies

(Source: Cassandra)

Evaluated and extracted the mean rating in the 'movies' database. The result is 8.76, representing the mean rating of the movie.

```
cqlsh:movie_database> SELECT * FROM movies WHERE title = 'Inception' ALLOW FILTERING;
```

movie_id	description	director	genre	rating	release_year	runtime	title
5f5c0a78-6c24-463c-888a-1c6ec6925840	A thief who enters the dreams of others to steal secrets from their subconscious.	Christopher Nolan	Sci-Fi	8.8	2010	148	Inception

(1 rows)
cqlsh:movie_database>

Figure 18: Retrieve movie details by title

(Source: Cassandra)

Retrieved movie attributes using the 'movies' table and the name of the movie is 'Inception'.

```
rmance. If you want to execute this query despite the performance unpredictability, use ALLOW FILTERING"
cqlsh:movie_database> SELECT * FROM movies WHERE release_year = 1994 ALLOW FILTERING;
```

movie_id	description	director	genre	rating	release_year	runtime	title
13359cf4-3918-450e-b9f9-cb4f55f2747c	It is based on the 1986 novel of the same name by Winston Groom and stars Tom Hanks, Robin Wright, Gary Sinise, Mykelti Williamson and Sally Field.	Robert Zemeckis	Drama	8.8	1994	142	Forrest Gump
982dfcce-4312-4a3c-908a-ee4b133202dc	A lion cub prince is tricked by a treacherous uncle into thinking he caused his father's death and flees into exile in despair, only to later accept in adulthood his identity and his responsibilities.	Roger Allers	Animation	8.5	1994	88	The Lion King
27fb9cc4-2d2f-4774-8e43-6e74592e02ce	The lives of two mob hitmen, a boxer, a gangster and his wife, and a pair of diner bandits intertwine in four tales of violence and redemption.	Quentin Tarantino	Crime	8.9	1994	154	Pulp Fiction
745f773f-e742-46b5-a217-ec0b4d408ce	The lives of two mob hitmen, a boxer, a gangster and his wife, and a pair of diner bandits intertwine in four tales of violence and redemption.	Quentin Tarantino	Crime	8.9	1994	154	Pulp Fiction
f63d9ce5-2510-4401-a165-2f1979d7c011	The presidencies of Kennedy and Johnson, the events of Vietnam, Watergate and other historical events unfold from the perspective of an Alabama man with an IQ of 75, whose only desire is to be reunited with his childhood sweetheart.	Robert Zemeckis	Drama	8.8	1994	142	Forrest Gump
b0c47f3a-d2de-4e2b-847e-7088ba160448	Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency.	Frank Darabont	Drama	9.3	1994	142	The Shawshank Redemption
e8fcdcc-c7b4-4f01-bb67-0edc0f58f32c	Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency.	Frank Darabont	Drama	9.3	1994	142	The Shawshank Redemption

(7 rows)
cqlsh:movie_database>

Figure 19: Retrieve movies released in a specific year

(Source: Cassandra)

Retrieved movie attributes using the 'movies' table and the release year is 1994.

```
cqlsh:movie_database> SELECT * FROM movies WHERE director = 'Christopher Nolan' ALLOW FILTERING;
```

movie_id	description	director	genre	rating	release_year	runtime	title
49cbde20-f8d8-45b9-9ff7-e6f8f56232cd	After a tragic accident, two stage magicians engage in a battle to create the ultimate illusion while sacrificing everything they have to outwit each other.	Christopher Nolan	Drama	8.5	2006	130	The Prestige
5f5c0a78-6c24-463c-888a-1c6ec6925840	A thief who enters the dreams of others to steal secrets from their subconscious.	Christopher Nolan	Sci-Fi	8.8	2010	148	Inception
4a69d456-4af2-4209-93f6-8c2b9dd5bf01	When the menace known as the Joker wreaks havoc and chaos on the people of Gotham, Batman must accept one of the greatest psychological and physical tests of his ability to fight injustice.	Christopher Nolan	Action	9	2008	152	The Dark Knight
60f6b85f-3662-4c92-b3a4-c968fa12bceb	A team of explorers travel through a wormhole in space in an attempt to ensure humanity's survival.	Christopher Nolan	Adventure	8.6	2014	169	Interstellar
20065c6f-5357-457e-9558-4b0ad446254a	"The Dark Knight" is a haunted film that leaps beyond its origins and becomes an engrossing tragedy. It creates characters we come to care about.	Christopher Nolan	Action	9	2008	172	The Dark Knight

(5 rows)
cqlsh:movie_database>

Figure 20: Retrieve movies directed by a specific director

(Source: Cassandra)

Retrieved movies using the 'movies' table and the director is 'Christopher Nolan'.

```
cqlsh:movie_database> SELECT * FROM movies WHERE rating > 8.5 ALLOW FILTERING;
```

movie_id	description	director	genre	rating	release_year	runtime
189	The Green Mile	Frank Darabont	Crime	8.6	1999	189
74022934-7029-404e-beee-16dfe8a9b90b	Gandalf and Aragorn lead the World of Men against Sauron's army to draw his gaze from Frodo and Sam as they approach Mount Doom with the One Ring.	Peter Jackson	Adventure	8.9	2003	201
3c19f301-36af-4a6f-9a10-787be9079814	While Frodo and Sam edge closer to Mordor with the help of the shifty Gollum, the divided fellowship makes a stand against Sauron's new ally, Saruman, and his hordes of Isengard.	Peter Jackson	Adventure	8.7	2002	179
13359cf4-3918-450e-b9f9-cb4f55f2747c	The Lord of the Rings: The Two Towers	Peter Jackson	Adventure	8.8	1994	142
57bed383-7a11-4606-ba62-6ad92625bf13	It is based on the 1986 novel of the same name by Winston Groom and stars Tom Hanks, Robin Wright, Gary Sinise, Mykelti Williamson and Sally Field.	Robert Zemeckis	Drama	8.8	1994	142
534f77db-086a-4ed1-8b6c-88b66304cf02	Forrest Gump	Robert Zemeckis	Drama	8.8	1994	142
274b9cc4-2d2f-4774-8e43-6e74592a02ce	The Godfather is set in the 1940s and takes place entirely within the world of the Corleones, a fictional New York Mafia family.	Francis Ford Coppola	Crime	9.2	1972	160
49cd0626-7808-43e9-977f-e0f87b222c0c	The Godfather	Francis Ford Coppola	Crime	9.2	1972	160
345530c-6daf-40b6-afa7-b273f9b28966	Following the Normandy Landings, a group of U.S. soldiers go behind enemy lines to retrieve a paratrooper whose brothers have been killed in action.	Steven Spielberg	Drama	8.6	1998	169
118	Saving Private Ryan	Steven Spielberg	Drama	8.6	1998	169
	The lives of two mob hitmen, a boxer, a gangster and his wife, and a pair of diner bandits intertwine in four tales of violence and redemption.	Quentin Tarantino	Crime	8.9	1994	154
	Pulp Fiction	Quentin Tarantino	Crime	8.9	1994	154
	A young F.B.I. cadet must receive the help of an incarcerated and manipulative cannibal killer to help catch another serial killer, a madman who skins his victims.	Jonathan Demme	Crime	8.6	1991	118
	The Silence of the Lambs	Jonathan Demme	Crime	8.6	1991	118

Figure 21: Retrieve movies with a rating higher than a certain value

(Source: Cassandra)

Retrieved the movies using the table of movies where the rating is greater than 8.5.

```
cqlsh:movie_database> SELECT * FROM movies WHERE genre = 'Drama' ALLOW FILTERING;
```

movie_id	description	director	genre	rating	release_year	runtime
13359cf4-3918-450e-b9f9-cb4f55f2747c	It is based on the 1986 novel of the same name by Winston Groom and stars Tom Hanks, Robin Wright, Gary Sinise, Mykelti Williamson and Sally Field.	Robert Zemeckis	Drama	8.8	1994	142
57bed383-7a11-4606-ba62-6ad92625bf13	Forrest Gump	Robert Zemeckis	Drama	8.8	1994	142
534f77db-086a-4ed1-8b6c-88b66304cf02	Following the Normandy Landings, a group of U.S. soldiers go behind enemy lines to retrieve a paratrooper whose brothers have been killed in action.	Steven Spielberg	Drama	8.6	1998	169
49cd0626-7808-43e9-977f-e0f87b222c0c	Saving Private Ryan	Steven Spielberg	Drama	8.6	1998	169
345530c-6daf-40b6-afa7-b273f9b28966	After a tragic accident, two stage magicians engage in a battle to create the ultimate illusion while sacrificing everything they have to outwit each other.	Christopher Nolan	Drama	8.5	2006	138
ff02a2c4-5b35-4531-9458-ba8902b1c312	The Prestige	Sidney Lumet	Drama	8.9	1957	95
12 Argy Men	Alcort attempts to prevent a miscarriage of justice by forcing his colleagues to reconsider the evidence.	Sidney Lumet	Drama	8.9	1957	95
24996607-7300-474a-ab57-1c79035be064	Fight Club	David Fincher	Drama	8.8	1999	139
463d9ce5-2510-4461-a165-2f1979d7c011	The Shawshank Redemption	Frank Darabont	Drama	9.3	1994	142
b0c47f3a-d2de-4e2b-847e-7088b4166448	The Shawshank Redemption	Frank Darabont	Drama	9.3	1994	142
8f0f0e0e-77d4-4691-bb67-0e0d0f58f37c	The Shawshank Redemption	Frank Darabont	Drama	9.3	1994	142

Figure 21: Retrieve movies within a specific genre

(Source: Cassandra)

Retrieved the movies within a specific genre, that is using where clause genre is equal to Drama.

```
cqlsh:movie_database> SELECT * FROM movies WHERE runtime <= 150 ALLOW FILTERING;
```

movie_id	description	director	genre	rating	release_year	runtime
13359cf4-3918-450e-b9f9-cb4f55f2747c	It is based on the 1986 novel of the same name by Winston Groom and stars Tom Hanks, Robin Wright, Gary Sinise, Mykelti Williamson and Sally Field.	Robert Zemeckis	Drama	8.8	1994	142
982dfcce-4312-4a3c-908a-ee4b133202dc	A lion cub prince is tricked by a treacherous uncle into thinking he caused his father's death and flees into exile in despair, only to later accept in adulthood his identity and his responsibilities.	Roger Allers	Animation	8.5	1994	88
49c0de20-f8d8-45b9-9ff7-e6f8f56232cd	After a tragic accident, two stage magicians engage in a battle to create the ultimate illusion while sacrificing everything they have to outwit each other.	Christopher Nolan	Drama	8.5	2005	130
34553c8c-9daf-40b6-afa7-b273f9b28966	A young F.B.I. cadet must receive the help of an incarcerated and manipulative cannibal killer to help catch another serial killer, a madman who skins his victims.	Jonathan Demme	Crime	8.6	1991	110
5f5c0a78-6c24-463c-888a-1c6ec6925840	A thief who enters the dreams of others to steal secrets from their subconscious.	Christopher Nolan	Sci-Fi	8.8	2010	148
ffd2aec4-5b35-4531-9458-ba8982b1c312	A jury holds a long deliberation to decide the fate of a man accused of murdering his wife.	Sidney Lumet	Drama	8.9	1957	96
24996607-7300-474a-b057-1c79035be064	Fight Club is a 1999 American film directed by David Fincher and starring Brad Pitt, Edward Norton, and Helena Bonham Carter.	David Fincher	Drama	8.8	1999	139
f63d9ce5-2510-4401-a165-2f1979d7c011	The presidencies of Kennedy and Johnson, the events of Vietnam, Watergate and other historical events unfold from the perspective of an Alabama man with an IQ of 75, whose only desire is to be reunited with his childhood sweetheart.	Robert Zemeckis	Drama	8.8	1994	142
b0c47f3a-d2de-4e2b-847e-7888b4160448	Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency.	Frank Darabont	Drama	9.3	1994	142
03d6544b-c84c-4991-96b0-87a71fdd7c68	Two detectives, a rookie and a veteran, hunt a serial killer who uses the seven deadly sins as his motives.	David Fincher	Crime	8.6	1995	127
e8fccddc-c7b4-4f01-bb67-0edc0f58f32c	Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency.	Frank Darabont	Drama	9.3	1994	142

Figure 22: Retrieve movies with a specific runtime or shorter

(Source: Cassandra)

Retrieved the movies using the table of movies using a where clause and the runtime is less than equal to 150.

```
cqlsh:movie_database> SELECT * FROM movies WHERE release_year >= 2000 AND release_year <= 2010 ALLOW FILTERING;
```

movie_id	description	director	genre	rating	release_year	runtime	title
748229b4-7d29-4d4e-beee-16df8a0bf0b	Gandalf and Aragorn lead the World of Men against Sauron's army to draw his gaze from Frodo and Sam as they approach Mount Doom with the One Ring.	Peter Jackson	Adventure	8.9	2003	201	The Lord of the Rings: The Return of the King
3c19f301-36af-4a6f-9a10-787be9079814	While Frodo and Sam edge closer to Mordor with the help of the shifty Gollum, the divided fellowship makes a stand against Sauron's new ally, Saruman, and his hordes of Isengard.	Peter Jackson	Adventure	8.7	2002	179	The Lord of the Rings: The Two Towers
49c0de20-f8d8-45b9-9ff7-e6f8f56232cd	After a tragic accident, two stage magicians engage in a battle to create the ultimate illusion while sacrificing everything they have to outwit each other.	Christopher Nolan	Drama	8.5	2006	130	The Prestige
5f5c0a78-6c24-463c-888a-1c6ec6925840	A thief who enters the dreams of others to steal secrets from their subconscious.	Christopher Nolan	Sci-Fi	8.8	2010	148	Inception
4a09c456-4af2-4309-93f6-8c2b9dd5bf01	When the menace known as the Joker wreaks havoc and chaos on the people of Gotham, Batman must accept one of the greatest psychological and physical tests of his ability to fight injustice.	Christopher Nolan	Action	9	2008	152	The Dark Knight
08206219-8730-4cd6-95b7-103ac112b40f	An undercover cop and a mole in the police attempt to identify each other while infiltrating an Irish gang in South Boston.	Martin Scorsese	Crime	8.5	2006	151	The Departed
73b8bf7f-9384-49f6-808a-7f9184df7219	A meek Hobbit from the Shire and eight companions set out on a journey to destroy the powerful One Ring and save Middle-earth from the Dark Lord Sauron.	Peter Jackson	Adventure	8.8	2001	178	The Lord of the Rings: The Fellowship of the Ring
75c66523-7055-41b9-a329-80414035ec44	A former Roman General sets out to exact vengeance against the corrupt emperor who murdered his family and sent him into slavery.	Ridley Scott	Action	8.5	2000	155	Gladiator
20065c6f-5357-457e-9558-4b0ad446254a	"The Dark Knight" is a haunted film that leaps beyond its origins and becomes an engrossing tragedy. It creates characters we come to care about.	Christopher Nolan	Action	9	2008	172	The Dark Knight

Figure 23: Retrieve movies released between two years

(Source: Cassandra)

Retrieved the movies that are released in two years between 2000 to 2010.

```
cqlsh:movie_database> SELECT * FROM movies WHERE director = 'Christopher Nolan' AND release_year > 2000 ALLOW FILTERING;
```

movie_id	description	director	genre	rating	release_year	runtime	title
49cbde20-f8d8-45b9-9ff7-e6f8f56232cd	After a tragic accident, two stage magicians engage in a battle to create the ultimate illusion while sacrificing everything they have to outwit each other.	Christopher Nolan	Drama	8.5	2006	130	The Prestige
5f5c0a78-6c24-463c-888a-1c6ec6925840	A thief who enters the dreams of others to steal secrets from their subconscious.	Christopher Nolan	Sci-Fi	8.8	2010	148	Inception
4a09c456-4af2-4309-93f6-8c2b9dd5bf01	When the menace known as the Joker wreaks havoc and chaos on the people of Gotham, Batman must accept one of the greatest psychological and physical tests of his ability to fight injustice.	Christopher Nolan	Action	9	2008	152	The Dark Knight
60f6b85f-36d2-4c92-b3a4-c968fa12bceb	A team of explorers travel through a wormhole in space in an attempt to ensure humanity's survival.	Christopher Nolan	Adventure	8.6	2014	169	Interstellar
20065c6f-5357-457e-9558-4b0ad446254a	"The Dark Knight" is a haunted film that leaps beyond its origins and becomes an engrossing tragedy. It creates characters we come to care about.	Christopher Nolan	Action	9	2008	172	The Dark Knight

(5 rows)

Figure 24: Retrieve movies directed by Christopher Nolan released after the year 2000

(Source: Cassandra)

Retrieved the movies where Christopher Nolan directs after 2000 year. Formed a database called “movie_database” using Cassandra and a table of “movies” with the attributes of movie_id, release_year, title, genre, director, rating, runtime, and description. 30 movie entries are added and 10 SELECT queries.

The rationale of design decisions

The procedure decisions are made to represent movies, containing variables efficiently (Kumar,2019). The data is organized in a method that allows for effortless return and analysis, allowing the queries with Cassandra.

Benefits and drawbacks

Benefits

The key necessities are efficient storage and scalability, retrieval, flexibility, and fault tolerance for the movie data system for different query functions.

Drawbacks

The system to handle complicated queries has limited capability are joins and collections where there are no limitations on the section keys.

Assignment 3

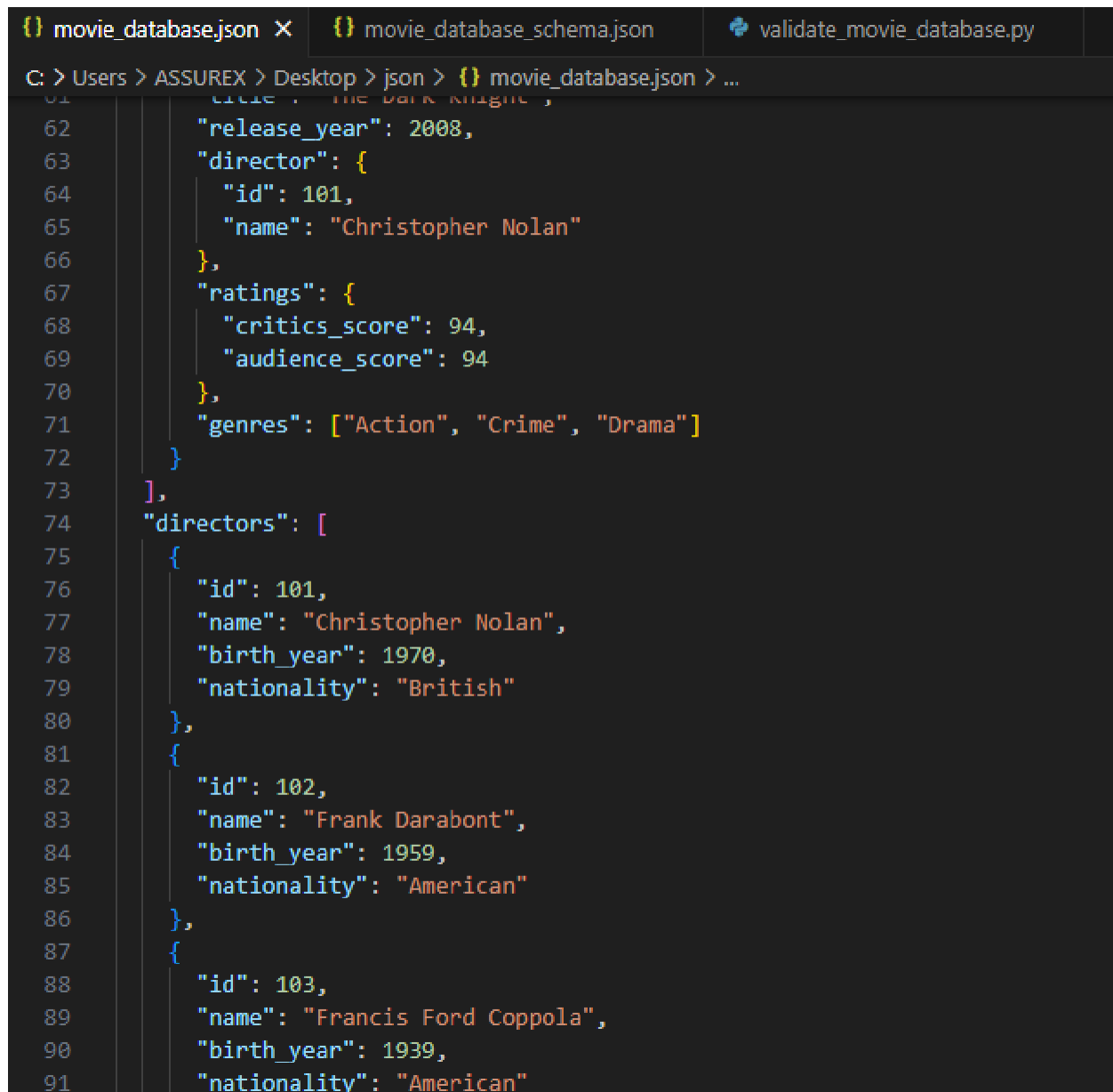
Implementation

```
1  {
2    "$schema": "http://json-schema.org/draft-07/schema#",
3    "type": "object",
4    "properties": {
5      "movies": {
6        "type": "array",
7        "items": {
8          "type": "object",
9          "properties": {
10             "id": { "type": "integer" },
11             "title": { "type": "string" },
12             "release_year": { "type": "integer" },
13             "director": {
14               "type": "object",
15               "properties": {
16                 "id": { "type": "integer" },
17                 "name": { "type": "string" }
18               },
19               "required": ["id", "name"]
20             },
21             "ratings": {
22               "type": "object",
23               "properties": {
24                 "critics_score": { "type": "integer" },
25                 "audience_score": { "type": "integer" }
26               },
27               "required": ["critics_score", "audience_score"]
28             },
29             "genres": { "type": "array", "items": { "type": "string" } }
30           },
31           "required": ["id", "title", "release_year", "director", "ratings", "genres"]
32         }
33     }
34 }
```

Figure 25: Movie database JSON schema

(Source: VS code)

The schema allows the efficient organization of data in a JSON configuration. The schema defines the design of a movie_database.



```
61  "title": "The Dark Knight",
62  "release_year": 2008,
63  "director": {
64    "id": 101,
65    "name": "Christopher Nolan"
66  },
67  "ratings": {
68    "critics_score": 94,
69    "audience_score": 94
70  },
71  "genres": ["Action", "Crime", "Drama"]
72  },
73  ],
74  "directors": [
75    {
76      "id": 101,
77      "name": "Christopher Nolan",
78      "birth_year": 1970,
79      "nationality": "British"
80    },
81    {
82      "id": 102,
83      "name": "Frank Darabont",
84      "birth_year": 1959,
85      "nationality": "American"
86    },
87    {
88      "id": 103,
89      "name": "Francis Ford Coppola",
90      "birth_year": 1939,
91      "nationality": "American"
```

Figure 26: Movie database JSON

(Source: VS code)

The data of JSON signifies a movie database record and this allows for organized storing with properties.

```

1 import json
2 from jsonschema import validate, ValidationError
3 def load_json(file_path):
4     with open(file_path, 'r') as file:
5         return json.load(file)
6 def validate_json(json_data, schema):
7     try:
8         validate(instance=json_data, schema=schema)
9         print("JSON document is valid against the schema.")
10    except ValidationError as e:
11        print(f"Validation Error: {e}")
12 def print_data_sections(json_data):
13     print("\nMovies:")
14     print(json.dumps(json_data.get('movies'), indent=2))
15
16     print("\nDirectors:")
17     print(json.dumps(json_data.get('directors'), indent=2))
18
19     print("\nRatings:")
20     print(json.dumps(json_data.get('ratings'), indent=2))
21
22     print("\nGenres:")
23     print(json.dumps(json_data.get('genres'), indent=2))
24 if __name__ == "__main__":
25     json_document = load_json("movie_database.json")
26     json_schema = load_json("movie_database_schema.json")
27     validate_json(json_document, json_schema)
28     print_data_sections(json_document)

```

Figure 27: Python code for checking JSON schema validation

(Source: VS code)

The program verifies the validity by utilizing the libraries of `json` and `jsonschema` of JSON data.

```
C:\Users\ASSUREX\Desktop\json>python validate_movie_database.py
JSON document is valid against the schema.
```

```
C:\Users\ASSUREX\Desktop\json>python validate_movie_database.py
JSON document is valid against the schema.
```

```
Movies:
```

```
[
  {
    "id": 1,
    "title": "Inception",
    "release_year": 2010,
    "director": {
      "id": 101,
      "name": "Christopher Nolan"
    },
    "ratings": {
      "critics_score": 92,
      "audience_score": 91
    },
    "genres": [
      "Action",
      "Sci-Fi",
      "Thriller"
    ]
  },
  {
    "id": 2,
    "title": "The Shawshank Redemption",
    "release_year": 1994,
    "director": {
      "id": 102,
      "name": "Frank Darabont"
    },
    "ratings": {
      "critics_score": 91,
      "audience_score": 98
    },
    "genres": [
      "Drama"
    ]
  },
  {

```

```

C:\Users\ASSUREX\Desktop\json>python validate_movie_database.py
JSON document is valid against the schema.

C:\Users\ASSUREX\Desktop\json>python validate_movie_database.py
JSON document is valid against the schema.

Movies:
[
  {
    "id": 1,
    "title": "Inception",
    "release_year": 2010,
    "director": {
      "id": 101,
      "name": "Christopher Nolan"
    },
    "ratings": {
      "critics_score": 92,
      "audience_score": 91
    },
    "genres": [
      "Action",
      "Sci-Fi",
      "Thriller"
    ]
  },
  {
    "id": 2,
    "title": "The Shawshank Redemption",
    "release_year": 1994,
    "director": {
      "id": 102,
      "name": "Frank Darabont"
    },
    "ratings": {
      "critics_score": 91,
      "audience_score": 98
    },
    "genres": [
      "Drama"
    ]
  },
  {

```

Figure 28: Output of validation of JSON schema

(Source: VS code)

The output shows the validity against the schema of the JSON document and it prints out data parts.

Rationale of design decisions

The relational structure between the movie database and entities is maintained by the JSON schema. Serialization and deserialization are easily made by the JSON schema and the storage of the database is allowed by the JSON schema integration. Data gathering into a hierarchical structure and efficient querying are manipulated by JSON schema.

Benefits and drawbacks

The flexibility of the database has been determined by the JSON schema implementation and data structure verification is also determined by the JSON schema integration with the movie database system (Krishanet *al.* 2019). Integration with API and web-based applications got easier by the JSON schema connection but the data integrity is very challenging for security purposes. Potential data redundancy and inconsistencies are included in the absence of enforced data validation. Complex types of queries are not so effective for this database and the data integrity is also very difficult to maintain.

Assignment 4

Implementation

```
>_MONGOSH  
  
> use movieDB  
< switched to db movieDB  
movieDB >
```

Figure 29: Create a database

(Source: Mongoddb)

The above figure shows the code to create a database in the MongoDB platform.

```
> db.createCollection("directors")  
< { ok: 1 }  
> db.createCollection("movies")  
< { ok: 1 }  
> db.createCollection("ratings")  
< { ok: 1 }  
> db.createCollection("genres")  
< { ok: 1 }  
movieDB >
```

Figure 30. Create four table

(Source: Mongoddb)

The above image shows the code to create tables named directors, movies, ratings, and genres by using the create collection method.

```
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('65e81c7510691bf15e4cc04d'),
    '1': ObjectId('65e81c7510691bf15e4cc04e'),
    '2': ObjectId('65e81c7510691bf15e4cc04f'),
    '3': ObjectId('65e81c7510691bf15e4cc050'),
    '4': ObjectId('65e81c7510691bf15e4cc051'),
    '5': ObjectId('65e81c7510691bf15e4cc052'),
    '6': ObjectId('65e81c7510691bf15e4cc053'),
    '7': ObjectId('65e81c7510691bf15e4cc054'),
    '8': ObjectId('65e81c7510691bf15e4cc055'),
```

Figure 31: Insert movie name

(Source: MongoDB)

The above image shows the insertion of the values for the movie table, the acknowledged is true and there are 20 different values inserted.

```
acknowledged: true,
insertedIds: {
  '0': ObjectId('65e81e0d10691bf15e4cc061'),
  '1': ObjectId('65e81e0d10691bf15e4cc062'),
  '2': ObjectId('65e81e0d10691bf15e4cc063'),
  '3': ObjectId('65e81e0d10691bf15e4cc064'),
  '4': ObjectId('65e81e0d10691bf15e4cc065'),
  '5': ObjectId('65e81e0d10691bf15e4cc066'),
  '6': ObjectId('65e81e0d10691bf15e4cc067'),
  '7': ObjectId('65e81e0d10691bf15e4cc068'),
  '8': ObjectId('65e81e0d10691bf15e4cc069'),
```

Figure 32: Insert director name

(Source: MongoDB)

Directors' names are inserted in this image, here the acknowledged value is true and 20 different values are inserted.

```

{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('65e820ba10691bf15e4cc075'),
    '1': ObjectId('65e820ba10691bf15e4cc076'),
    '2': ObjectId('65e820ba10691bf15e4cc077'),
    '3': ObjectId('65e820ba10691bf15e4cc078'),
    '4': ObjectId('65e820ba10691bf15e4cc079'),
    '5': ObjectId('65e820ba10691bf15e4cc07a'),
    '6': ObjectId('65e820ba10691bf15e4cc07b'),
    '7': ObjectId('65e820ba10691bf15e4cc07c'),
  }
}

```

Figure 33: Insert ratings

(Source: MongoDB)

The ratings are inserted in this image, here the acknowledged value is true and 20 different values are inserted.

```

{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('65e8211710691bf15e4cc089'),
    '1': ObjectId('65e8211710691bf15e4cc08a'),
    '2': ObjectId('65e8211710691bf15e4cc08b'),
    '3': ObjectId('65e8211710691bf15e4cc08c'),
    '4': ObjectId('65e8211710691bf15e4cc08d'),
    '5': ObjectId('65e8211710691bf15e4cc08e'),
    '6': ObjectId('65e8211710691bf15e4cc08f'),
  }
}

```

Figure 34: Insert genres

(Source: MongoDB)

The genres are inserted in this image, here the acknowledged value is true and 20 different values are inserted.

```

> db.movies.find({ director: "Steven Spielberg" })
< {
  _id: ObjectId('65e81c7510691bf15e4cc059'),
  title: 'Saving Private Ryan',
  director: 'Steven Spielberg',
  release_year: 1998,
  budget: 700000000
}
{
  _id: ObjectId('65e81c7510691bf15e4cc05a'),
  title: 'Schindler's List',
  director: 'Steven Spielberg',
  release_year: 1993,
  budget: 220000000
}

```

Figure 35: Find all movies directed by Steven Spielberg

(Source: MongoDB)

The above image shows the query for finding all movies which are directed by Steven Spielberg and here all the movie's names are displayed accordingly.

```

> db.ratings.aggregate([
  { $lookup: { from: "movies", localField: "movie", foreignField: "title", as: "movie_info" } },
  { $match: { "movie_info.release_year": { $gt: 2000 } } },
  { $group: { _id: null, avg_rating: { $avg: "$rating" } } }
])
< {
  _id: null,
  avg_rating: 8.0125
}

```

Figure 36: Find the average rating of movies released after 2000

(Source: MongoDB)

Find the average ratings of movies which are released after the year 2000 is shown in the figure where the value of id is null and the value of average rating is 8.01.

```

> db.genres.find({ genre: "Action" })
< {
  _id: ObjectId('65e8211710691bf15e4cc089'),
  movie: 'The Dark Knight',
  genre: 'Action'
}
{
  _id: ObjectId('65e8211710691bf15e4cc08c'),
  movie: 'Kill Bill: Vol. 1',
  genre: 'Action'
}

```

Figure 37: Find all movies with the "Action" genre

(Source: MongoDB)

The above image shows the code for find all the movies name which are from the genre action where all the action genre movie names are displayed.

```

> db.directors.find({ awards: { $all: ["Academy Award", "Academy Award"] } })
< {
  _id: ObjectId('65e81e0d10691bf15e4cc061'),
  name: 'Steven Spielberg',
  nationality: 'American',
  dob: '1946-12-18',
  awards: [
    'Academy Award',
    'Golden Globe'
  ]
}

```

Figure 38: Find the directors who have won more than one Academy Award

(Source: MongoDB)

The directors who have won the Academy Award more than one time are shown in the figure.

```
> db.ratings.aggregate([
  { $sort: { rating: -1 } },
  { $limit: 3 }
])
< {
  _id: ObjectId('65e820ba10691bf15e4cc07c'),
  movie: 'The Shawshank Redemption',
  rating: 9.3,
  voters: 2433456
}
{
  _id: ObjectId('65e820ba10691bf15e4cc07d'),
  movie: 'The Godfather',
  rating: 9.2,
  voters: 1977862
}
{
  _id: ObjectId('65e820ba10691bf15e4cc075'),
  movie: 'The Dark Knight',
  rating: 9,
  voters: 2314189
}
```

Figure 39: Find the top 3 highest rated movies

(Source: MongoDB)

The above image shows the top three top-rated movies 'The Shawshank Redemption', 'the godfather' and 'the dark knight' are shown in the above figure.

```

> db.genres.aggregate([
  { $lookup: { from: "ratings", localField: "movie", foreignField: "movie", as: "ratings_info" } },
  { $unwind: "$ratings_info" },
  { $group: { _id: "$genre", total_voters: { $sum: "$ratings_info.voters" } } }
])
< {
  _id: 'Crime',
  total_voters: 5916129
}
{
  _id: 'War',
  total_voters: 1832490
}
{
  _id: 'Western',
  total_voters: 1407694
}
{
  _id: 'Mystery',
  total_voters: 1185392
}
{

```



```

{
  _id: 'Biography',
  total_voters: 1576210
}
{
  _id: 'Action',
  total_voters: 7897602
}
{
  _id: 'Adventure',
  total_voters: 1734567
}
{
  _id: 'Romance',
  total_voters: 2209876
}
{
  _id: 'Sci-Fi',
  total_voters: 3929178
}

```

Figure 40: Find the total number of voters for each genre

(Source: MongoDB)

The total number of voters of each genre is shown in the above image where for crime total voters are 5916129, for war total voters, are 1832499, for western total voters are 1407694, mystery total voters, are 1185392.

```

> db.ratings.aggregate([
  { $sort: { voters: -1 } },
  { $limit: 1 }
])
< {
  _id: ObjectId('65e820ba10691bf15e4cc07c'),
  movie: 'The Shawshank Redemption',
  rating: 9.3,
  voters: 2433456
}

```

Figure 41: Find the movie with the highest number of voters

(Source: MongoDB)

The above image shows the movie name which contains the highest number of voters where the movie name is 'The Shawshank redemption', the rating is 9.3 and the voter's number is 2433456.

```
> db.movies.aggregate([
  { $lookup: { from: "directors", localField: "director", foreignField: "name", as: "director_info" } },
  { $match: { "director_info.nationality": "American" } },
  { $group: { _id: null, avg_budget: { $avg: "$budget" } } }
])
< {
  _id: null,
  avg_budget: 57437500
}
```

Figure 26: Find the average budget of movies directed by American directors

(Source: MongoDB)

The average budget of those movies which is directed by American directors is shown in the figure where the average budget is 57437500.

```
> db.genres.aggregate([
  { $match: { genre: "Crime" } },
  { $lookup: { from: "movies", localField: "movie", foreignField: "title", as: "movies_info" } },
  { $unwind: "$movies_info" },
  { $group: { _id: null, total_budget: { $sum: "$movies_info.budget" } } }
])
< {
  _id: null,
  total_budget: 111000000
}
```

Figure 42: Find the total budget spent on movies in the "Crime" genre

(Source: MongoDB)

The total budget spent on the crime genre movie is shown in the above image where the total budget is 111000000.

```

> db.movies.aggregate([
  { $lookup: { from: "directors", localField: "director", foreignField: "name", as: "director_info" } },
  { $match: { "director_info.nationality": "British" } },
  { $lookup: { from: "ratings", localField: "title", foreignField: "movie", as: "ratings_info" } },
  { $unwind: "$ratings_info" },
  { $group: { _id: null, avg_rating: { $avg: "$ratings_info.rating" } } }
])
< {
  _id: null,
  avg_rating: 8.8
}

```

Figure 43: Find the average rating of movies directed by British directors

(Source: MongoDB)

The average ratings of those movies which are directed by British directors are shown in the figure where the average rating is 8.8.

The rationale of design decisions

In the above query the find function and aggregation method are used and here all the data are retrieved using this (Diogoet al. 2019). Group and lookup are mainly used for data manipulation and finding queries are used to filter data based on specific conditions and the power of MongoDB's querying capacities.

Benefits and drawbacks

The benefits of using MongoDB queries depend on their flexibility and implementation. They deliver strong aggregation capacities, permitting complicated data manipulations.

However, MongoDB's query language needs to be less reflexive for beginners, and complex queries need a more in-depth understanding of its syntax (Čerešňákand Kvet, 2019). Further, specific operations such as joins are less efficient compared to relational databases.

Assignment 5

Implementation

```
1 CREATE (m1:Movie {movie_id: 2, name: 'The Matrix', genre: 'Action',
2   release_year: 1999})
3 CREATE (m2:Movie {movie_id: 5, name: 'The Shawshank Redemption', genre:
4   'Drama', release_year: 1994})
5 CREATE (m3:Movie {movie_id: 1, name: 'Pulp Fiction', genre: 'Crime',
6   release_year: 1994})
7 CREATE (m4:Movie {movie_id: 8, name: 'The Dark Knight', genre: 'Action',
8   release_year: 2008})
9 CREATE (m5:Movie {movie_id: 21, name: 'Inception', genre: 'Sci-Fi',
10  release_year: 2010})
11 CREATE (m6:Movie {movie_id: 55, name: 'The Godfather', genre: 'Crime',
12  release_year: 1972})
```

Added 10 labels, created 10 nodes, set 40 properties, completed after 44 ms.

Figure 44: Creating Movie table and inserting value in this
(Source: neo4j)

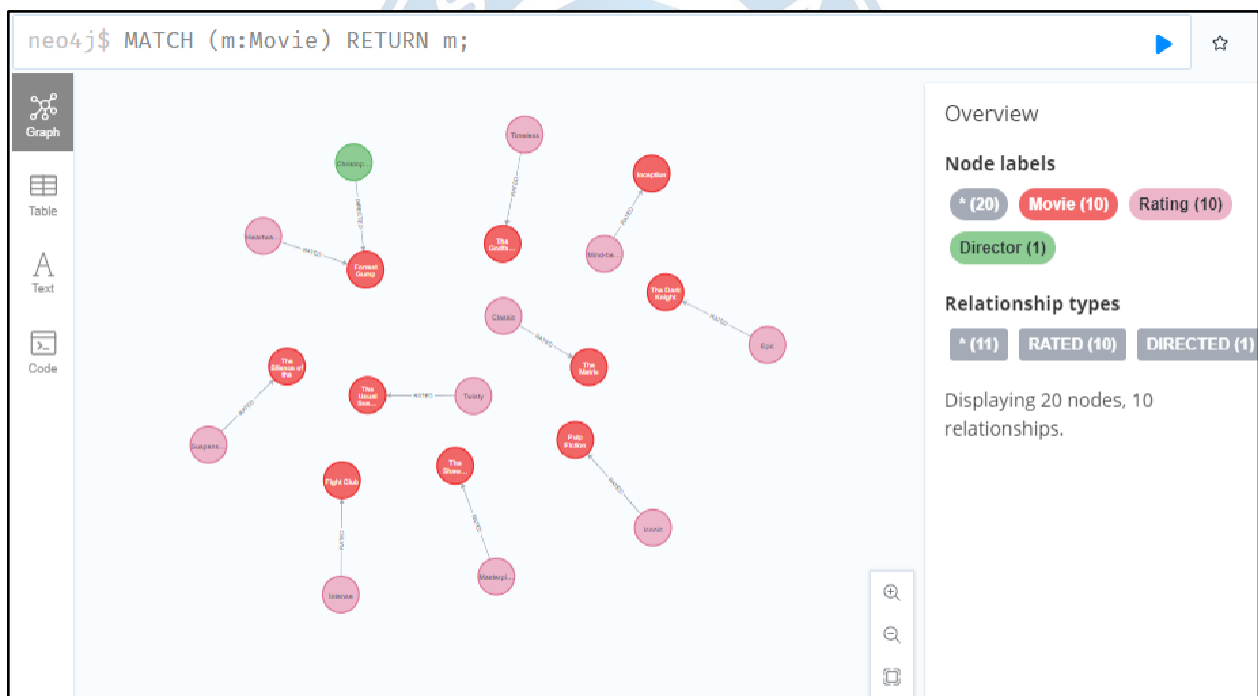


Figure 45: Graphical show for movie table data
(Source: neo4j)

```
1 CREATE (d1:Director {director_id: 51, name: 'Lana Wachowski', age: 56,
nationality: 'American'})
2 CREATE (d2:Director {director_id: 103, name: 'Frank Darabont', age: 62,
nationality: 'American'})
3 CREATE (d3:Director {director_id: 42, name: 'Quentin Tarantino', age: 58,
nationality: 'American'})
4 CREATE (d4:Director {director_id: 62, name: 'Christopher Nolan', age: 51,
nationality: 'British'})
5 CREATE (d5:Director {director_id: 121, name: 'Francis Ford Coppola', age:
82, nationality: 'American'})
6 CREATE (d6:Director {director_id: 63, name: 'Robert Zemeckis', age: 69,
nationality: 'American'})
```

Added 10 labels, created 10 nodes, set 40 properties, completed after 36 ms.

Figure 46: Creating Director Table and inserting value in this
(Source: neo4j)

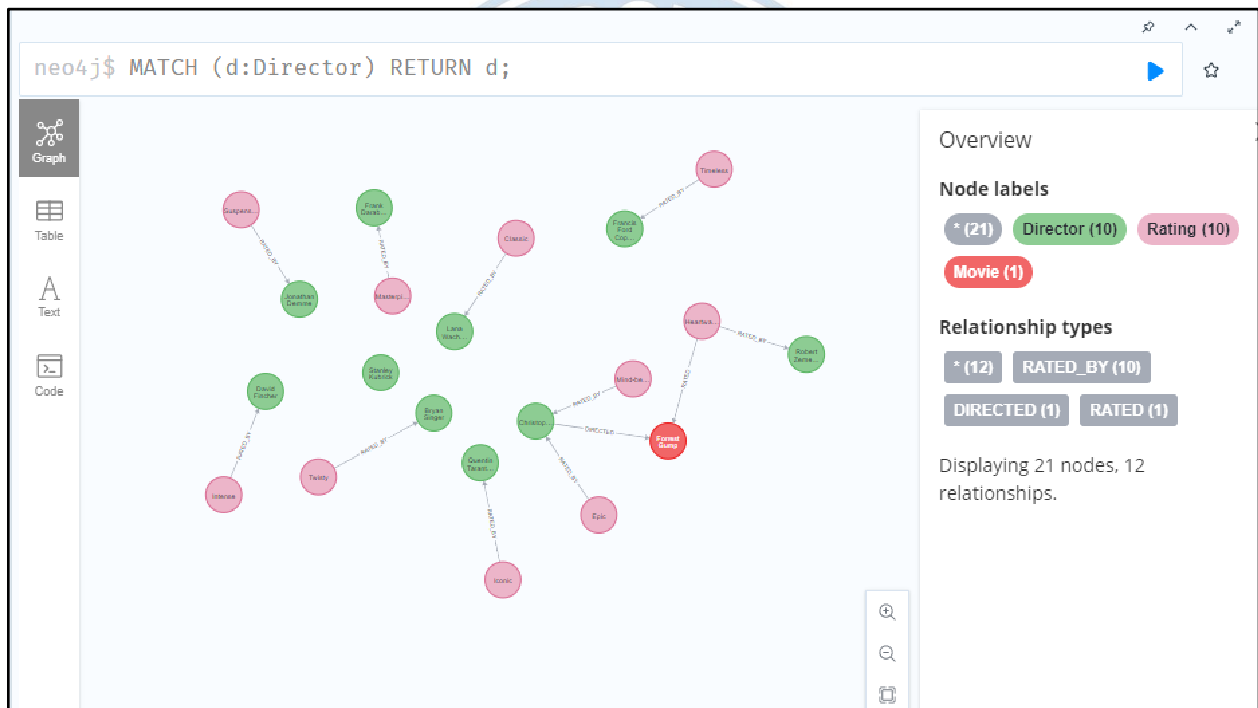


Figure 47: Graphical show for director table data
(Source: neo4j)

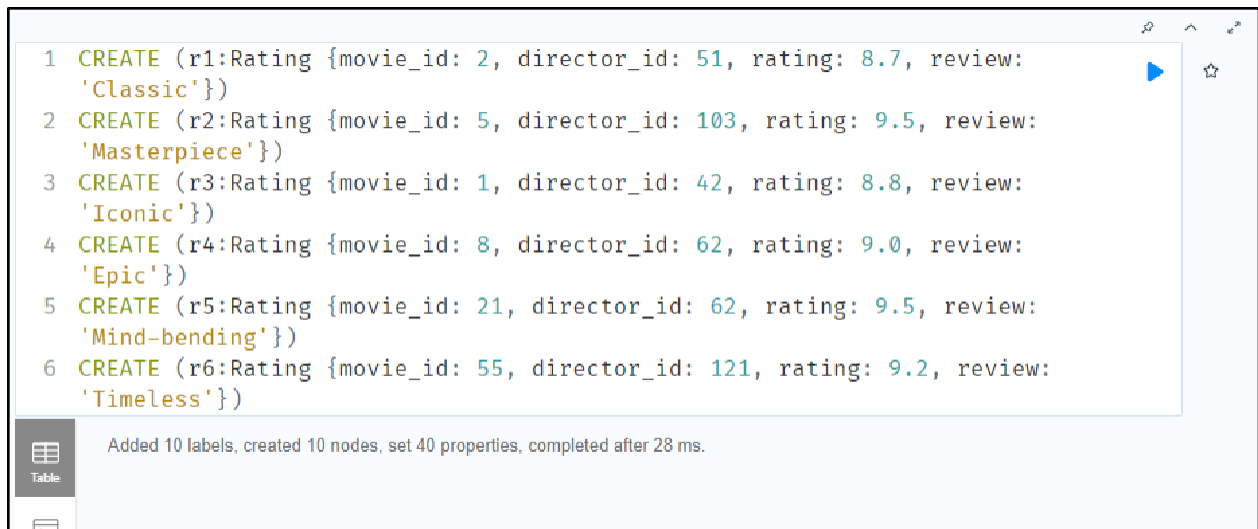


Figure 48: Creating Rating table and inserting value in this

(Source: neo4j)

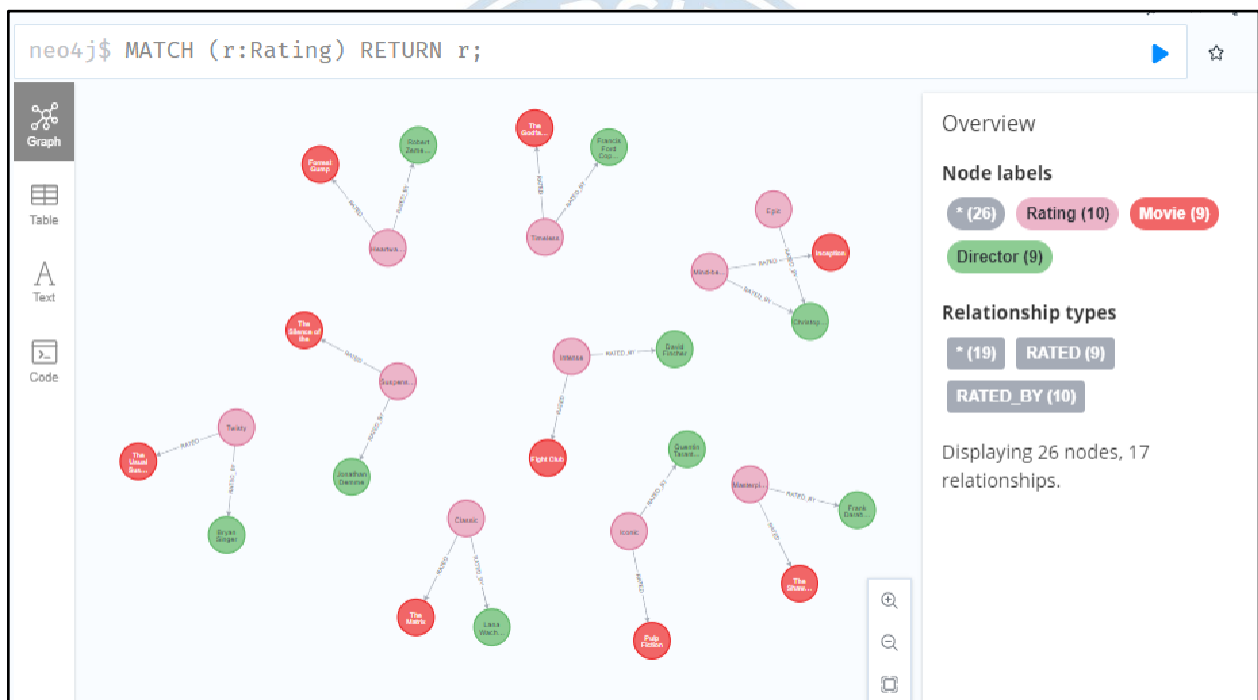


Figure 49: Graphical show for rating table data

(Source: neo4j)

```

1 // Connecting Movies and Directors
2 MATCH (m:Movie), (d:Director)
3 WHERE m.movie_id = d.director_id
4 CREATE (d)-[:DIRECTED]→(m);
5
6 // Connecting Movies and Ratings
7 MATCH (m:Movie), (r:Rating)
8 WHERE m.movie_id = r.movie_id
9 CREATE (r)-[:RATED]→(m);
10
11 // Connecting Directors and Ratings
12 MATCH (d:Director), (r:Rating)

```

```

movie$ MATCH (m:Movie), (d:Director) WHERE m.movie_id = d.director_id CREATE ...
movie$ MATCH (m:Movie), (r:Rating) WHERE m.movie_id = r.movie_id CREATE (r)-[...
movie$ MATCH (d:Director), (r:Rating) WHERE d.director_id = r.director_id CRE...

```

Figure 50: Connecting tables

(Source: neo4j)

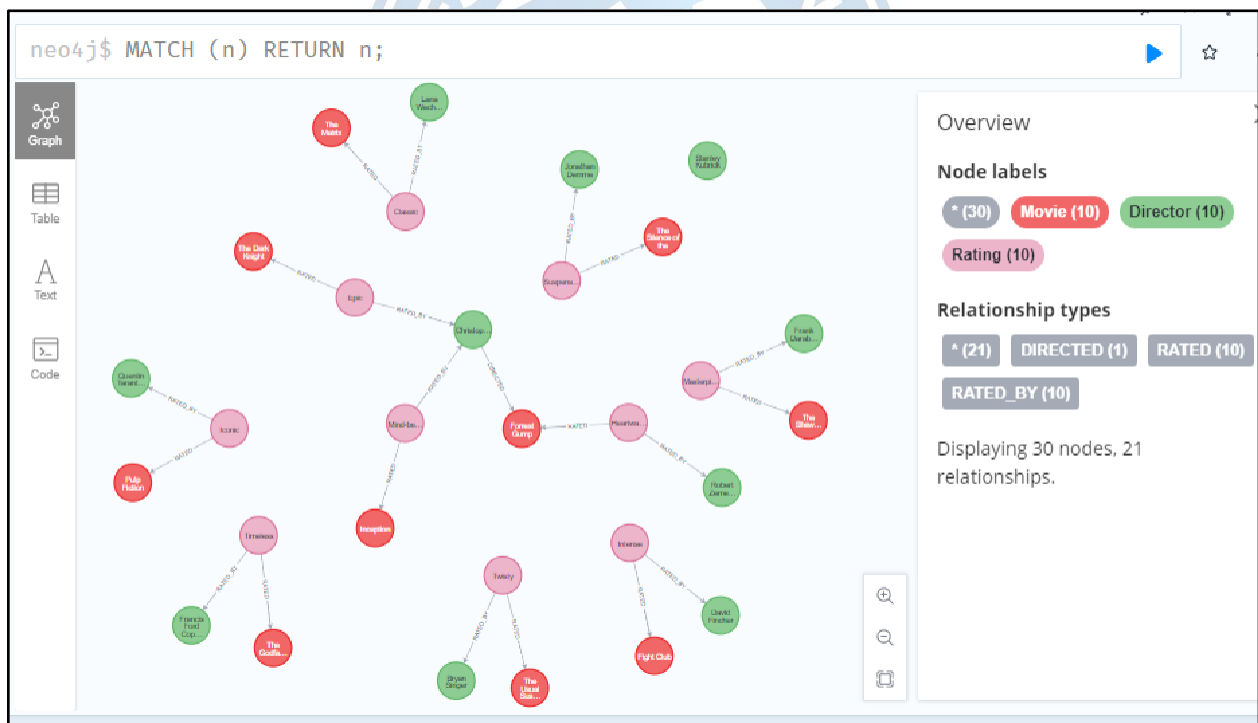


Figure 51: Show the entire graph

(Source: neo4j)

The above image shows the entire graph, here the red nodes denote movie names, green nodes denote director names and the pink node denotes ratings and each node contains 10 values.

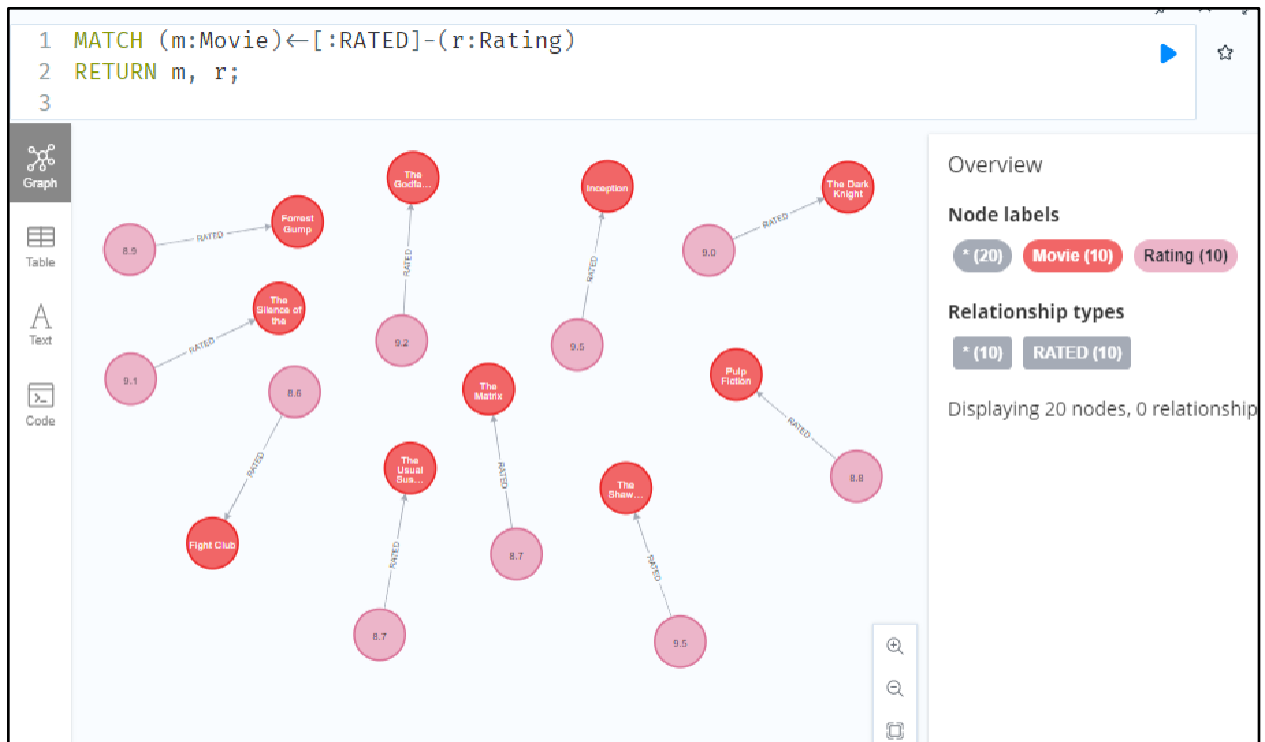


Figure 52: Show Movies and their Ratings

(Source: neo4j)

The above image shows the movie's names and their ratings, here the red nodes denote movie names, and the pink node denotes ratings and each node contains 10 values.

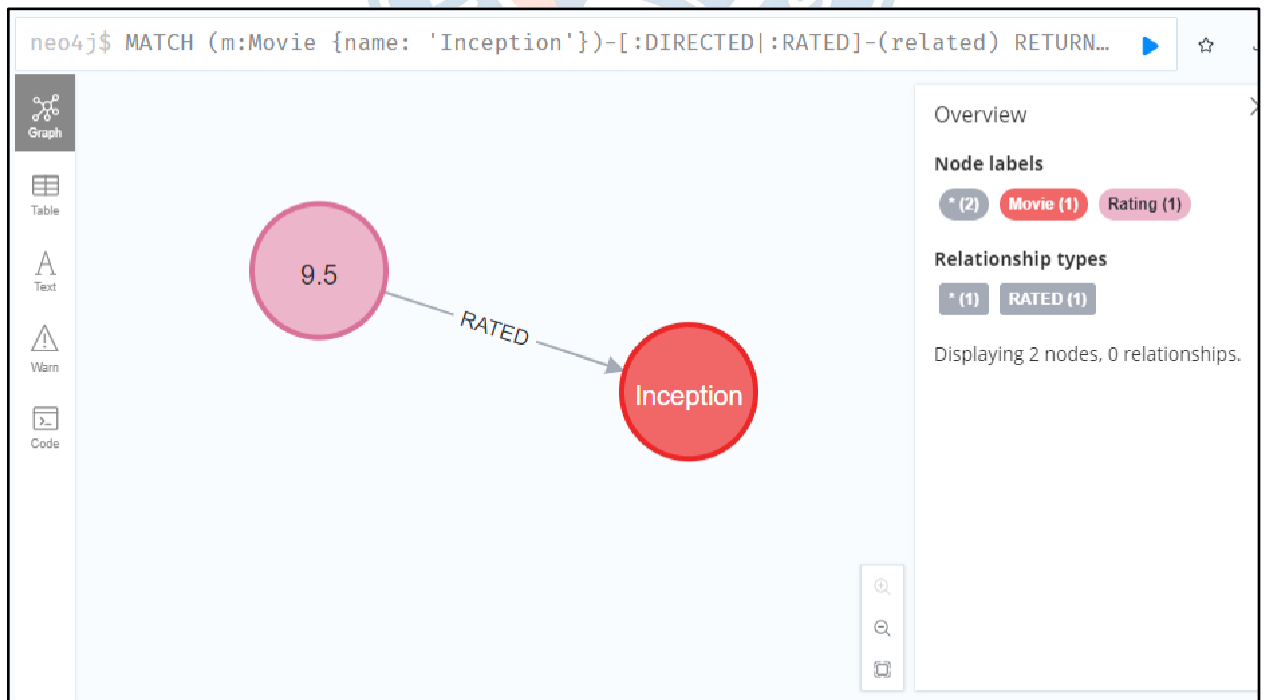
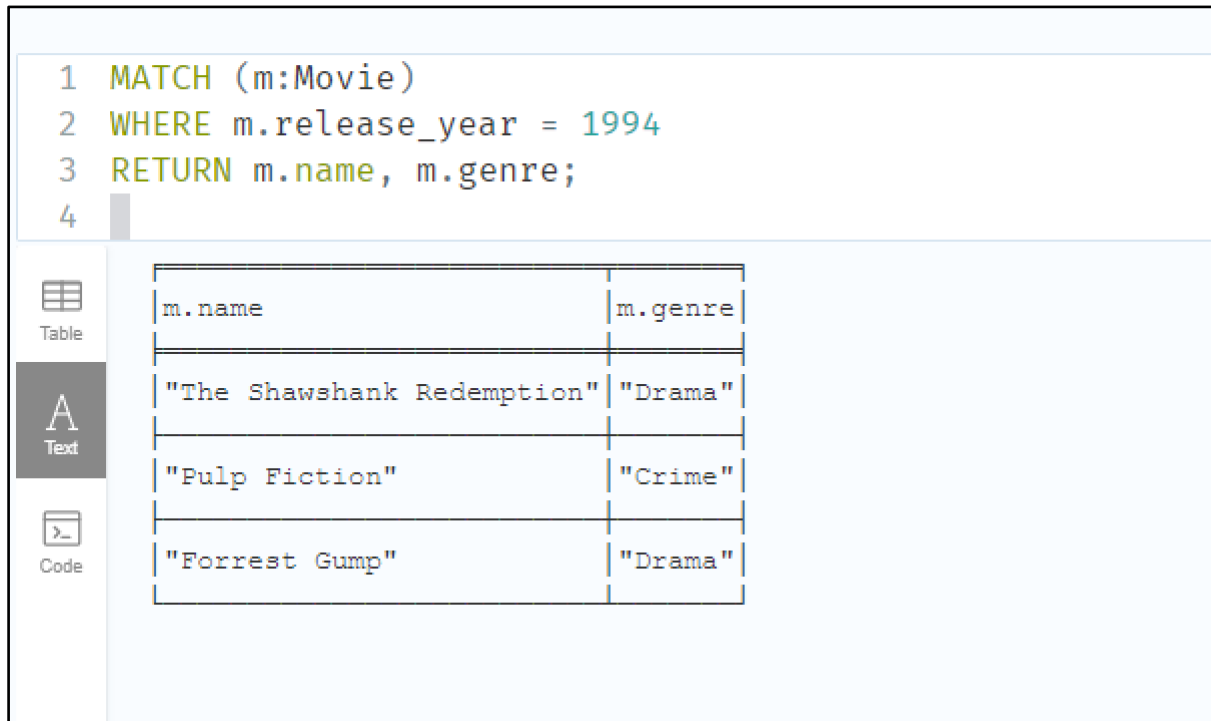


Figure 53: Show the graph around a specific Movie (e.g., 'Inception')

(Source: neo4j)

The above image shows the graph that contains a specific movie name inception and the rating for this is 9.5.



The screenshot shows a Neo4j interface with a Cypher query and its results. The query is as follows:

```
1 MATCH (m:Movie)
2 WHERE m.release_year = 1994
3 RETURN m.name, m.genre;
4
```

On the left, there are three view options: Table, Text (which is selected), and Code. The results are displayed in a table with two columns: m.name and m.genre.

m.name	m.genre
"The Shawshank Redemption"	"Drama"
"Pulp Fiction"	"Crime"
"Forrest Gump"	"Drama"

Figure 54: Showing movies released in 1994

(Source: neo4j)

The above image shows the movie name which was released in 1994 and here the output shows the three movie names with genre names.

```
1 MATCH (m:Movie)
2 WHERE m.genre = 'Action'
3 RETURN m.name, m.release_year;
4
```

Table

Text

Code

m.name	m.release_year
"The Matrix"	1999
"The Dark Knight"	2008
"The Matrix"	1999
"The Dark Knight"	2008

Figure 55: Find movies with a specific genre: Action

(Source: neo4j)

The above image shows the names of the movies that are from the action genre, here the movie names are the matrix and The Dark Knight. And the release dates are 1999 and 2008.

neo4j\$ MATCH (m:Movie)←[:RATED]-(r:Rating) WITH m, r ORDER BY r.rating DESC W...

movie_name	m.genre	rating
"The Shawshank Redemption"	"Drama"	9.5
"Inception"	"Sci-Fi"	9.5
"The Godfather"	"Crime"	9.2
"The Silence of the Lambs"	"Thriller"	9.1
"The Dark Knight"	"Action"	9.0
"Forrest Gump"	"Drama"	8.9
"Pulp Fiction"	"Crime"	8.8
"The Matrix"	"Action"	8.7
"The Usual Suspects"	"Mystery"	8.7
"Fight Club"	"Drama"	8.6

Figure 56: Movies with the highest ratings for each Genre

(Source: neo4j)

The above image shows the name of the movie with the highest ratings for each genre, here the 10 movie names are shown with the genre name and with the ratings.

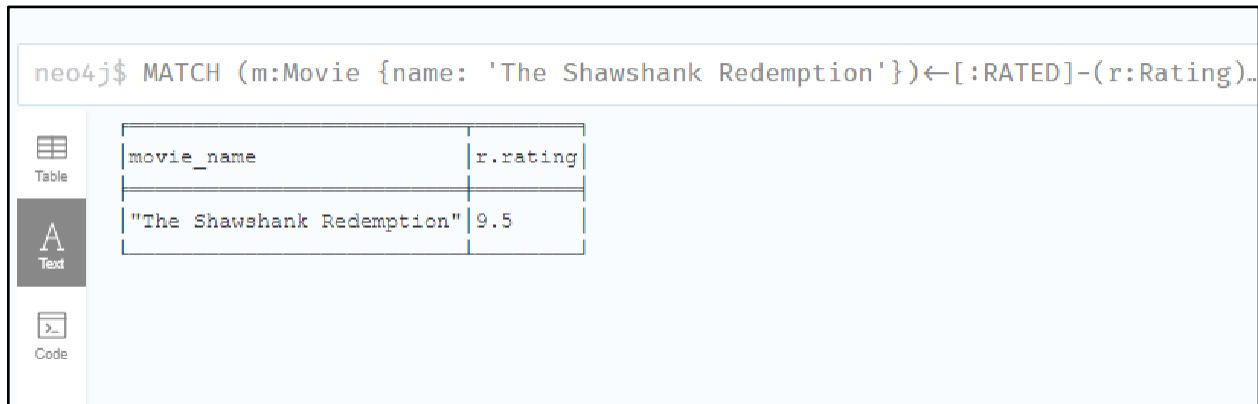
neo4j\$ MATCH (m:Movie)←[:RATED]-(r:Rating) RETURN m.genre AS genre, AVG(r.rat..

genre	average_rating
"Sci-Fi"	9.5
"Thriller"	9.1
"Drama"	9.0
"Crime"	9.0
"Action"	8.85
"Mystery"	8.7

Figure 57: Show the average rating for each Genre

(Source: neo4j)

The above image shows the average ratings for the average genre name where the genres are sci-fi, drama, crime, action, and mystery, and the ratings are 9.5, 9.1, 9.9, 8.085 and 9.7 respectively.



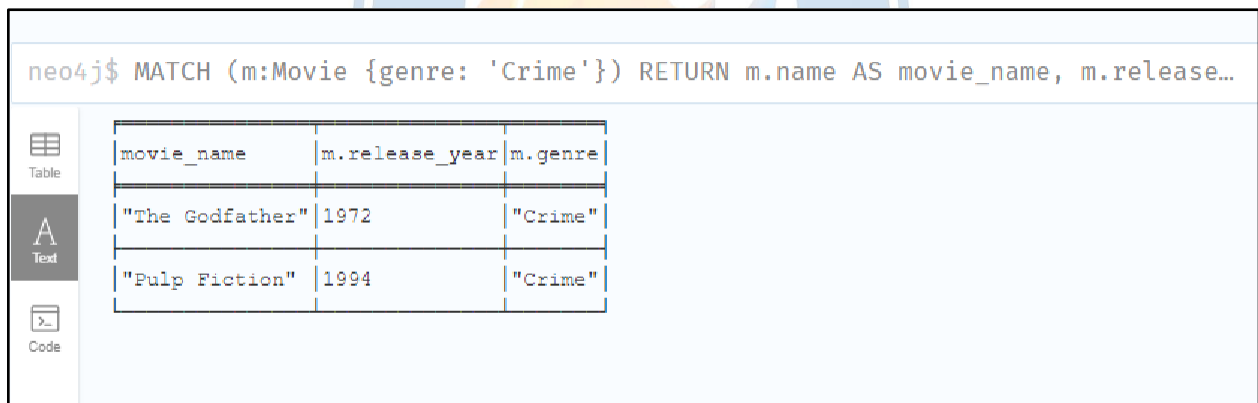
The image shows a Neo4j interface with a Cypher query and its result. The query is: `neo4j$ MATCH (m:Movie {name: 'The Shawshank Redemption'})←[:RATED]-(r:Rating)...`. The result is displayed in a table with two columns: `movie_name` and `r.rating`. The table contains one row with the movie name and its rating.

movie_name	r.rating
"The Shawshank Redemption"	9.5

Figure 58: Show the Distribution of Ratings for 'The Shawshank Redemption'

(Source: neo4j)

The distribution of ratings for the movie The Shawshank Redemption is shown in the figure where the rating is 9.5.



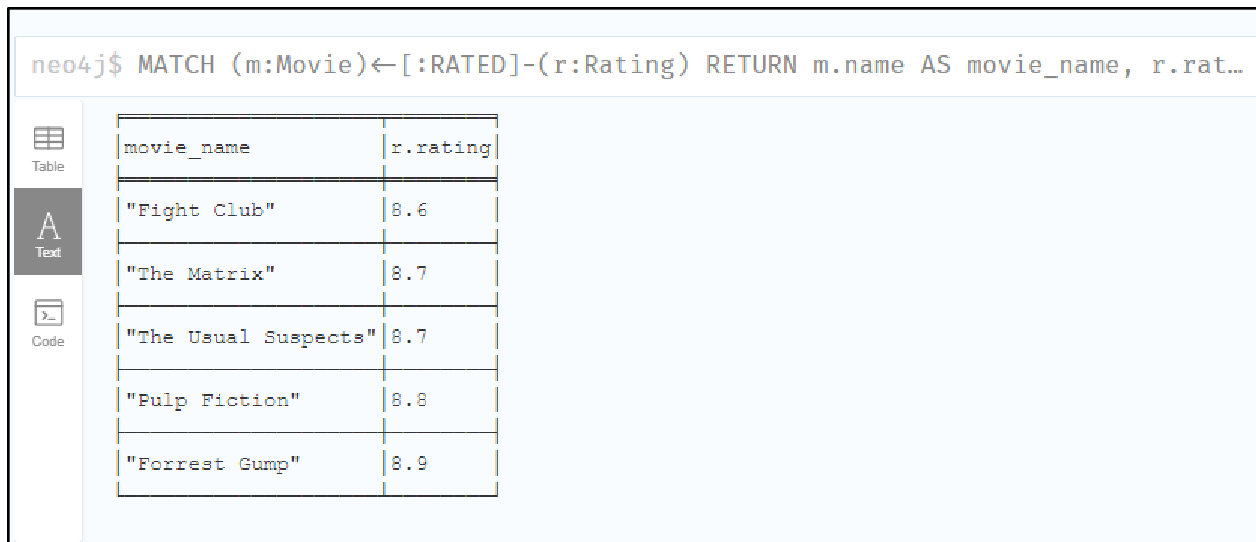
The image shows a Neo4j interface with a Cypher query and its result. The query is: `neo4j$ MATCH (m:Movie {genre: 'Crime'}) RETURN m.name AS movie_name, m.release...`. The result is displayed in a table with three columns: `movie_name`, `m.release_year`, and `m.genre`. The table contains two rows of crime movies.

movie_name	m.release_year	m.genre
"The Godfather"	1972	"Crime"
"Pulp Fiction"	1994	"Crime"

Figure 59: Showing crime movies

(Source: neo4j)

The above image shows the names of the crime movies where the movie names are The Godfather and Pulp Fiction and the release dates are 1972 and 1994 respectively.



The screenshot shows the Neo4j web interface. At the top, a Cypher query is entered: `neo4j$ MATCH (m:Movie)←[:RATED]-(r:Rating) RETURN m.name AS movie_name, r.rat...`. Below the query, a table displays the results. The table has two columns: 'movie_name' and 'r.rating'. The results list five movies: 'Fight Club' (8.6), 'The Matrix' (8.7), 'The Usual Suspects' (8.7), 'Pulp Fiction' (8.8), and 'Forrest Gump' (8.9). On the left side of the interface, there are three icons: a table icon labeled 'Table', a text icon labeled 'Text', and a code icon labeled 'Code'. The 'Table' icon is currently selected.

movie_name	r.rating
"Fight Club"	8.6
"The Matrix"	8.7
"The Usual Suspects"	8.7
"Pulp Fiction"	8.8
"Forrest Gump"	8.9

Figure 60: Show 5 low-rated movies in the movie table

(Source: neo4j)

5 low-rated movies are shown in the above figure, here the movie names are Fight Club, the Matrix, the usual suspects, Pulp Fiction, and ForrestGump.

The rationale of design decisions

A complex relationship between the entities of the database has been represented by the database made in the Neo4j platform. Common genre identification among the movie types and traversal connection establishment are made by the graph database (Islamet *al.* 2021). A natural fit has been made in Neo4j for the connection between movie names and movie directors.

Benefits and drawbacks

Different complex relationships are handled very easily in the Neo4j platform and the relationship between movie genre and movie rating has been established in the Neo4j platform (Parmarand Roy, 2018). User-friendly graph visualization is made for the flexibility of the implemented movie database and the connection between entities is shown by the graph database.

Storage requirement is a very big drawback in the relational database made in the Neo4j platform. Graphs and charts of large sizes are very difficult to optimize in this platform and maintaining scalability in distributed environments are also very challenging.

References

- Ali, A., Naeem, S., Anam, S. and Ahmed, M.M., 2023. A state of art survey for big data processing and nosql database architecture. *International Journal of Computing and Digital Systems*, 14(1), pp.1-1.
- Besta, M., Gerstenberger, R., Peter, E., Fischer, M., Podstawski, M., Barthels, C., Alonso, G. and Hoefler, T., 2023. Demystifying graph databases: Analysis and taxonomy of data organization, system designs, and graph queries. *ACM Computing Surveys*, 56(2), pp.1-40.
- Bonifati, A., Holubová, I., Prat-Pérez, A. and Sakr, S., 2020. Graph generators: State of the art and open challenges. *ACM computing surveys (CSUR)*, 53(2), pp.1-30.
- databases. *Transportation Research Procedia*, 40, pp.170-177.
- Debrouvier, A., Parodi, E., Perazzo, M., Soliani, V. and Vaisman, A., 2021. A model and query language for temporal graph databases. *The VLDB Journal*, 30(5), pp.825-858.
- databases. *Future Internet*, 11(2), p.43.
- Freitas, S., Dong, Y., Neil, J. and Chau, D.H., 2020. A large-scale database for graph representation learning. *arXiv preprint arXiv:2011.07682*.
- Islam, S., Hasan, K. and Shahriyar, R., 2021. Mining developer questions about major NoSQL databases. *Int. J. Comput. Appl*, 975, p.8887.
- Krishan, K., Gupta, G. and Bhathal, G.S., 2023. A REVIEW AND COMPARISON OF KEY DISTRIBUTED DATABASE CHARACTERISTICS ACROSS SEVERAL NOSQL DISTRIBUTED DATABASES. *Journal of Data Acquisition and Processing*, 38(1), p.321.
- Kumar, A., 2019. *Implementing real time recommendation systems using graph algorithms & exploring graph analytics in a graph database platform (neo4j)* (Doctoral dissertation, Dublin Business School).

Parmar, R.R. and Roy, S., 2018. MongoDB as an efficient graph database: An application of document oriented NOSQL database. In *Data Intensive Computing Applications for Big Data* (pp. 331-358). IOS Press.

Sicari, S., Rizzardi, A. and Coen-Porisini, A., 2022. Security&privacy issues and challenges in NoSQL databases. *Computer Networks*, 206, p.108828.

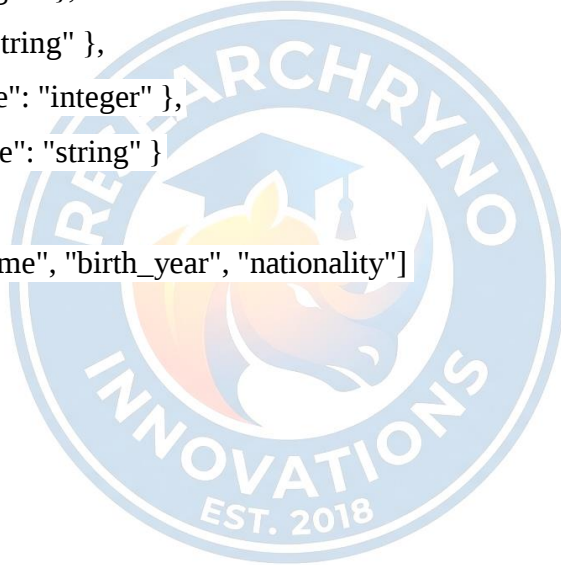


Appendix

JSON schema

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "properties": {
    "movies": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "id": { "type": "integer" },
          "title": { "type": "string" },
          "release_year": { "type": "integer" },
          "director": {
            "type": "object",
            "properties": {
              "id": { "type": "integer" },
              "name": { "type": "string" }
            },
            "required": ["id", "name"]
          },
          "ratings": {
            "type": "object",
            "properties": {
              "critics_score": { "type": "integer" },
              "audience_score": { "type": "integer" }
            },
            "required": ["critics_score", "audience_score"]
          }
        }
      }
    }
  }
}
```

```
    "genres": { "type": "array", "items": { "type": "string" } }  
  },  
  "required": ["id", "title", "release_year", "director", "ratings", "genres"]  
}  
,  
"directors": {  
  "type": "array",  
  "items": {  
    "type": "object",  
    "properties": {  
      "id": { "type": "integer" },  
      "name": { "type": "string" },  
      "birth_year": { "type": "integer" },  
      "nationality": { "type": "string" }  
    },  
    "required": ["id", "name", "birth_year", "nationality"]  
  },  
},  
"ratings": {  
  "type": "array",  
  "items": {  
    "type": "object",  
    "properties": {  
      "movie_id": { "type": "integer" },  
      "source": { "type": "string" },  
      "score": { "type": "number" }  
    },  
    "required": ["movie_id", "source", "score"]  
  },  
},  
},
```



```
"genres": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "id": { "type": "integer" },
      "name": { "type": "string" }
    },
    "required": ["id", "name"]
  }
},
"required": ["movies", "directors", "ratings", "genres"]
}
```

JSON document

```
{
  "movies": [
    {
      "id": 1,
      "title": "Inception",
      "release_year": 2010,
      "director": {
        "id": 101,
        "name": "Christopher Nolan"
      },
      "ratings": {
        "critics_score": 92,
        "audience_score": 91
      },
    },
  ],
}
```

```
"genres": ["Action", "Sci-Fi", "Thriller"]
```

```
},
```

```
{
```

```
  "id": 2,
```

```
  "title": "The Shawshank Redemption",
```

```
  "release_year": 1994,
```

```
  "director": {
```

```
    "id": 102,
```

```
    "name": "Frank Darabont"
```

```
  },
```

```
  "ratings": {
```

```
    "critics_score": 91,
```

```
    "audience_score": 98
```

```
  },
```

```
  "genres": ["Drama"]
```

```
},
```

```
{
```

```
  "id": 3,
```

```
  "title": "The Godfather",
```

```
  "release_year": 1972,
```

```
  "director": {
```

```
    "id": 103,
```

```
    "name": "Francis Ford Coppola"
```

```
  },
```

```
  "ratings": {
```

```
    "critics_score": 97,
```

```
    "audience_score": 98
```

```
  },
```

```
  "genres": ["Crime", "Drama"]
```

```
},
```



```
{
  "id": 4,
  "title": "Pulp Fiction",
  "release_year": 1994,
  "director": {
    "id": 104,
    "name": "Quentin Tarantino"
  },
  "ratings": {
    "critics_score": 94,
    "audience_score": 96
  },
  "genres": ["Crime", "Drama"]
},
{
  "id": 5,
  "title": "The Dark Knight",
  "release_year": 2008,
  "director": {
    "id": 101,
    "name": "Christopher Nolan"
  },
  "ratings": {
    "critics_score": 94,
    "audience_score": 94
  },
  "genres": ["Action", "Crime", "Drama"]
}
],
"directors": [
```



```
{
  "id": 101,
  "name": "Christopher Nolan",
  "birth_year": 1970,
  "nationality": "British"
},
{
  "id": 102,
  "name": "Frank Darabont",
  "birth_year": 1959,
  "nationality": "American"
},
{
  "id": 103,
  "name": "Francis Ford Coppola",
  "birth_year": 1939,
  "nationality": "American"
},
{
  "id": 104,
  "name": "Quentin Tarantino",
  "birth_year": 1963,
  "nationality": "American"
},
{
  "id": 105,
  "name": "Steven Spielberg",
  "birth_year": 1946,
  "nationality": "American"
}
```



```
],  
"ratings": [  
  {  
    "movie_id": 1,  
    "source": "IMDb",  
    "score": 8.8  
  },  
  {  
    "movie_id": 2,  
    "source": "IMDb",  
    "score": 9.3  
  },  
  {  
    "movie_id": 3,  
    "source": "IMDb",  
    "score": 9.2  
  },  
  {  
    "movie_id": 4,  
    "source": "IMDb",  
    "score": 8.9  
  },  
  {  
    "movie_id": 5,  
    "source": "IMDb",  
    "score": 9.0  
  }  
],  
"genres": [  
  {
```



```
"id": 1,  
  "name": "Action"  
},  
{  
  "id": 2,  
  "name": "Sci-Fi"  
},  
{  
  "id": 3,  
  "name": "Thriller"  
},  
{  
  "id": 4,  
  "name": "Drama"  
},  
{  
  "id": 5,  
  "name": "Crime"  
}  
]  
}
```

