Group Project - Neo4j

of Systems and Methods for Big and Unstructured Data Course
(SMBUD)
held by
Brambilla Marco
Tocchetti Andrea

Group 78

Pisante Giuseppe	Raffaelli Martina
10696936	10709893

Academic year 2024/2025

POLITECNICO
MILANO 1863

Contents

1 Introduction

The project aims to design and implement a database system to support the management of data related to the film industry. The database will include entities such as Person (actor, director, writer), Title (film, TV series), Episode, ratings, and genre. The goal is to create a comprehensive system that can store and query information about films, TV series, and the people involved in their production. The project will develop with Neo4j, to exploit the relations between the entities and find significant insights from the data, such as collaborations between directors and actors, trend of genres over time, most working actors and the most successful films.

2 Assumptions

The project is based on the following assumptions:

- Each person has a unique ID and a name, surname, and date of birth. Some could also have a death date, if passed away.
- Each person can be associated with multiple roles (actor, director, writer, archive footage, music department, producer)
- Each title has a unique identifier and a title type (film, TV series, shortfilm)
- Each title can have multiple episodes
- Each title can have multiple genres
- Each user can rate a title if and only if it has watched the film
- Each title has a unique rating, average rating from the users, and the number of votes
- Each person can be associated with multiple titles
- Each title can have multiple people associated with it

3 ER diagram

Figure 1: E-R Diagram

3.1 Entities

Starting from the considerations previously exposed regarding the implementation hypotheses, we have drawn an ER diagram (**Figure 1**) which includes 5 different entities and 7 many-to-many relationships described below in the logical model:

- **Person**(nconst, PrimaryName, BirthYear, DeathYear, PrimaryProfession, KnownForTitles)
- **Title**(tconst, PrimaryTitle, OriginalTitle, TitleType, StartYear, EndYear, RuntimeMinutes, Genres)
- **Episode**(tconst, ParentTconst, SeasonNumber, EpisodeNumber)
- **Ratings**(tconst, AverageRating, NumVotes)

- **Genre**(Name)

The **Person** entity describes every possible individual with their own personal data, including their primary profession and titles they are known for. The **Title** entity represents films, TV series or short films with their respective attributes. The **Episode** entity is used to detail episodes of TV series, linked to their parent series. The **Ratings** entity captures the average rating and number of votes for each title. Finally, the **Genre** entity categorizes the titles into different genres.

3.2 Relationships

ACTED_IN $(Person) - [: ACTED\_IN] - > (Title)$
Relationship between a Person, whose primary profession is actor, and a Title.

DIRECTED $(Person) - [: DIRECTED] - > (Title)$
Relationship between a Person, whose primary profession is director, and a Title.

WROTE $(Person) - [: WROTE] - > (Title)$
Relationship between a Person, whose primary profession is writer, and a Title.

PART_OF $(Episode) - [: PART\_OF] - > (Title)$
Relationship between an Episode and its parent Title, whose TitleType is TV series.

HAS_GENRE $(Title) - [: HAS\_GENRE] - > (Genre)$
Relationship between a Title and a Genre.

HAS_RATING $(Title) - [: HAS\_RATING] - > (Rating)$
Relationship between a Title and its Rating.

3.3 Constraints:

Ciao

4 Cypher Queries

4.1 Market Analysis for Marketing Department

This section of the Cypher Queries aims at providing the Marketing department with insights on the share within the market by providing the number of orders in the Countries in which the Company operates. In particular, this is done by providing an overview on the total orders per country, the revenue per country, a demographic overview per country and an analysis of the most present brands within a specific category, which in this study was chosen as the category with the most orders for further relevance.

4.1.1 Total orders by Country

```
MATCH (u:User)-[:PLACES]->(o:Order)
RETURN u.country AS country, COUNT(o) AS total_orders
ORDER BY total_orders DESC;
```

Table 1: Total Orders by Country

Country	Total Orders
China	42,986
United States	28,099
Brasil	18,262
South Korea	6,620
France	5,968
United Kingdom	5,673
Germany	5,286
Spain	4,965
Japan	2,945
Australia	2,630
Belgium	1,441
Poland	325
Colombia	19
España	4
Austria	2
Deutschland	1

4.1.2 Revenue per Country

```
MATCH (u:User)-[:PLACES]->(o:Order)-[:CONTAINS]->(oi:OrderItem)
MATCH (oi)-[:REFERS_TO]->(p:Product)
RETURN u.country AS country, SUM(p.cost) AS total_revenue
ORDER BY total_revenue DESC;
```

Table 2: Total Revenue by Country

Country	Total Revenue
China	1,800,865.85
United States	1,162,263.11
Brasil	747,519.90
South Korea	278,958.29
France	243,130.02
United Kingdom	242,263.07
Germany	217,875.61
Spain	210,278.08
Japan	124,807.02
Australia	106,009.74
Belgium	60,226.06
Poland	13,386.90
Colombia	572.51
España	91.87
Deutschland	65.70
Austria	41.51

4.1.3 Demographic Overview

```
MATCH (u:User)
RETURN u.country AS country,
     COUNT(u) AS total_users,
     AVG(u.age) AS average_age
ORDER BY total_users DESC;
```

4.1.4 Categories with the most orders

```
MATCH (oi:OrderItem)-[:REFERS_TO]->(p:Product)
RETURN p.product_category AS category, COUNT(oi) AS total_orders
ORDER BY total_orders DESC
LIMIT 3;
```

4.1.5 Most present brands within the intimates category

```
MATCH (oi:OrderItem)-[:REFERS_TO]->(p:Product)
```

Table 3: Total Users and Average Age by Country

Country	Total Orders	Average Value
China	34,150	40.89
United States	22,522	41.21
Brasil	14,507	41.19
South Korea	5,316	41.25
France	4,700	41.57
United Kingdom	4,561	41.05
Germany	4,155	40.86
Spain	4,062	41.01
Japan	2,438	40.89
Australia	2,146	40.98
Belgium	1,185	39.54
Poland	235	42.43
Colombia	17	34.88
Deutschland	2	40.50
España	2	38.50
Austria	2	50.00

Table 4: Categories with the most orders

Category	Total Orders
Intimates	13,474
Jeans	12,698
Tops & Tees	11,925

```
WHERE p.category = "Intimates"

RETURN p.brand AS brand,
      COUNT(oi) AS total_sales
ORDER BY total_sales DESC
LIMIT 5;
```

4.1.6 Most present brands within the intimates category per country

```
MATCH (u:User)-[:ORDERED]->(oi:OrderItem)-[:REFERS_TO]->(p:Product)
```

Table 5: Most present brands within the intimates categor

Brand	Total Sales
Bali	405
Maidenform	383
Hanes	364
Laura	342
Vanity Fair	306

```
WHERE p.category = "Intimates"
WITH u.country AS country,
    p.brand AS brand,
    COUNT(oi) AS total_sales
ORDER BY country, total_sales DESC
WITH country, COLLECT({brand: brand, total_sales: total_sales}) AS brand_sal
RETURN country,
      brand_sales[0].brand AS top_brand,
      brand_sales[0].total_sales AS top_sales
ORDER BY country;
```

4.2 Logistic Analysis for Logistics Department

This section of the Cypher Queries aims at providing the Logistics department with insights on the real-time tracking of the orders and all the possible information related to tuser in order to make the delivery as smooth as possible. In particular, this is done by evaluating the closest distribution center for a specific user, the status of the order, the update of the history of the orders of user and to check if the user has some order pending, in which case the items can be sent together.

Table 6: Total Orders and Average Value by Country

Country	**Total Orders**	**Average Value**
China	34,150	40.89
United States	22,522	41.21
Brasil	14,507	41.19
South Korea	5,316	41.25
France	4,700	41.57
United Kingdom	4,561	41.05
Germany	4,155	40.86
Spain	4,062	41.01
Japan	2,438	40.89
Australia	2,146	40.98
Belgium	1,185	39.54
Poland	235	42.43
Colombia	17	34.88
Deutschland	2	40.50
España	2	38.50
Austria	2	50.00

5 References & Sources

[1] Course Slides

[2] https://pysimplegui.readthedocs.io/en/latest/call

[3] https://py2neo.org/

[4] https://neo4j.com/docs/cypher-manual/current/

[5] https://neo4j.com/developer/python/

[6] http://iniball.altervista.org/Software/ProgER

[7] https://neo4j.com/developer/cypher/

[8] https://pandas.pydata.org/docs/