RAPPORT DE PROJET

Base de données Netflix

AGBO Merveille
BOUSLAH Zohra
DIOP Aida
FAITAI Marie-Ruth
LUWEH ADJIM NGARTI Exaucé
ORY Maxime
TOMILINA Ekaterina

UE Bases de données
Responsable de l'UE : DIALLO Gayo **Année 2020-2021**

Introduction **3**

Choix et réalisation d'une base de données graphe **4**

Schéma relationnel des données **7**

Requêtes et analyse **8**

Quels sont les pays avec le plus de contenu? 8

Comment ont évolué les films face aux séries? 8

Quelles sont les catégories prédominantes? 9

Quel est le rapport entre le nombre de films et l'année de sortie? 9

La durée des films présente-t-elle une particularité? 10

Qu'en est-il des séries? 10

Quels sont les acteurs ayant joué dans le plus de films? 11

Conclusion **11**

Répartition des tâches **11**

Introduction

Depuis 2010, la quantité de séries télévisées sur Netflix a pratiquement triplé, comme le montre un rapport de Flixable rédigé en 2018. De plus, le nombre de films a diminué de 2000. Nous nous sommes donc intéressés dans notre étude à la base de données concernant les films et séries disponibles sur Netflix en 2019. Nous avions les informations suivantes : l'identifiant du film/série, le titre, le directeur, les acteurs principaux, le pays, la date d'ajout, l'année de sortie, la restriction d'âge, la durée, la catégorie et la description.
Nous avons choisi d'utiliser une base de données graphe, choix que nous allons tout d'abord justifier. Ensuite, nous présenterons le schéma relationnel des données, et enfin exécuterons quelques requêtes dont nous effectuerons l'analyse.

Choix et réalisation d'une base de données graphe

Nous avions le choix pour ce projet entre quatre types de bases de données : graphe, colonne, clé-valeur, ou document. Pour traiter nos données, nous avons choisi la première.
En effet:

- Nous avions besoin d'une conservation des relations entre les données ainsi que d'effectuer des requêtes concernant le contenu : une base clé-valeur était donc à exclure.
- Nos données n'étaient pas des documents d'où l'élimination d'une base documents
- Nous avions besoin de requêtes concernants des données précises, une base de données orientée colonnes aurait nécessité trop de colonnes à consulter pour chaque requête
- Ainsi, le stockage dans une base de données graphe nous convient parfaitement. On a une vue d'ensemble des données et des relations entre elles, sans obligation d'utiliser énormément de jointures lors d'une requête.

Pour effectuer ces graphes, nous avons utilisé le logiciel Neo4j sandbox ainsi que la plateforme Jupyter. Nous avons importé nos données dans Jupyter à l'aide du code Python suivant, en ajoutant une colonne spécifique concernant la durée de chaque film (en minutes) et de chaque série (en saisons) en utilisant la librairie py2neo. L'usage de Python nous a permis de travailler tous ensemble assez facilement car il s'agit d'un langage connu de tous les membres. Nous avons effectué les requêtes avec le langage Cypher.

```
# load the data
df = pd.read_csv('netflix_titles.csv')
df["date_added"] = pd.to_datetime(df['date_added'])
df['year'] = df['date_added'].dt.year
df['month'] = df['date_added'].dt.month
df['day'] = df['date_added'].dt.day

df['duration_film'] = df['duration']
df['duration_serie'] = df['duration']

for i in range(len(df['duration'])):
    if "min" in df['duration'][i]:
        df['duration_film'][i]=int(df['duration'][i].replace(" min",""))
    else :
        df['duration_film'][i]= 0

for i in range(len(df['duration'])):
    if "min" in df['duration'][i]:
        df['duration_serie'][i]=0
    elif "Season" in df['duration'][i]:
        df['duration_serie'][i] = (df['duration'][i].replace(" Season",""))
    else :
        df['duration_serie'][i] = (df['duration'][i].replace("s",""))

df.head()
```

Ensuite, nous avons synchronisé nos actions sur Jupyter avec Neo4j sandbox par le biais du code suivant :

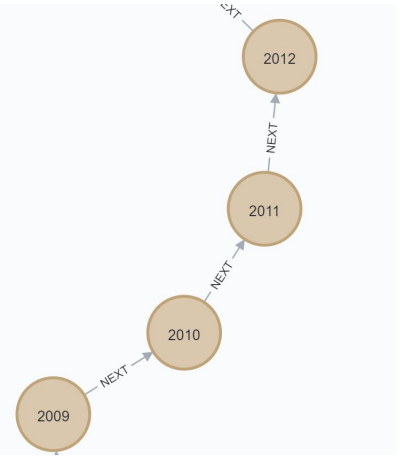
```
#Connect to neo4j sandbox
ip="54.226.9.151"
port="32904"
pwd="provision-centimeters-airplanes"
graph = Graph("bolt://"+ip+":"+port, auth=("neo4j", pwd))
```

Nous avons créé les nœuds représentant les films, acteurs, directeurs, dates de sortie… ci-dessous le code utilisé concernant les films :

```
request ="""
CREATE (m:Movie {id: {show_id},title:{title}})
SET
m.director = {director},
m.country = {country},
m.date_str = {date_added},
m.release_year = {release_year},
m.rating = {rating},
m.duration = {duration} ,
m.listed_in = {listed_in} ,
m.description = {description},
m.cast={cast},
m.year = {year},
m.month = {month},
m.day = {day},
m.type = {type_movie},
m.duration_film = {duration_film},
m.duration_serie = {duration_serie};
"""
```

Nous avons également mis en place des relations entre les nœuds : par exemple, les nœuds représentant les années sont reliés dans le sens du temps grâce au code suivant qui crée la relation NEXT.

```
request="""
MATCH (year:Year)
WITH year
ORDER BY year.value
WITH collect(year) AS years
FOREACH(i in RANGE(0, size(years)-2) |
    FOREACH(year1 in [years[i]] |
        FOREACH(year2 in [years[i+1]] |
            MERGE (year1)-[:NEXT]->(year2))));
"""
run_request(request,LOAD_DATA=True)
```

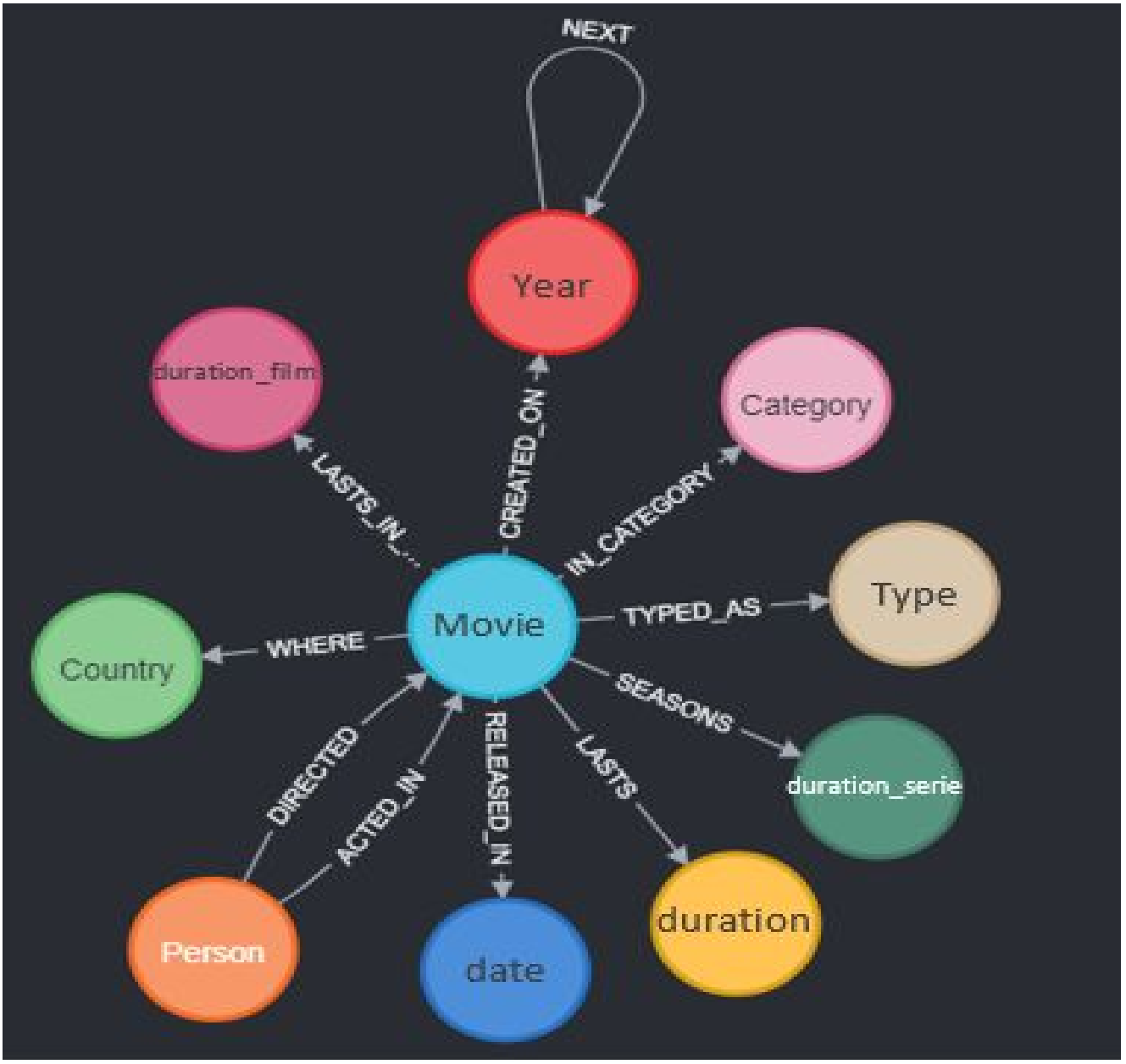


Un autre exemple, voici la commande qui crée la relation WHERE qui indique le lieu de tournage d'un film ou d'une série.

```
request ="""
MATCH (m:Movie)
WHERE m.country IS NOT NULL
MERGE (c:Country {name: trim(m.country)})
MERGE (m)-[:WHERE]->(c);
"""
run_request(request,LOAD_DATA=True)
```

Schéma relationnel des données

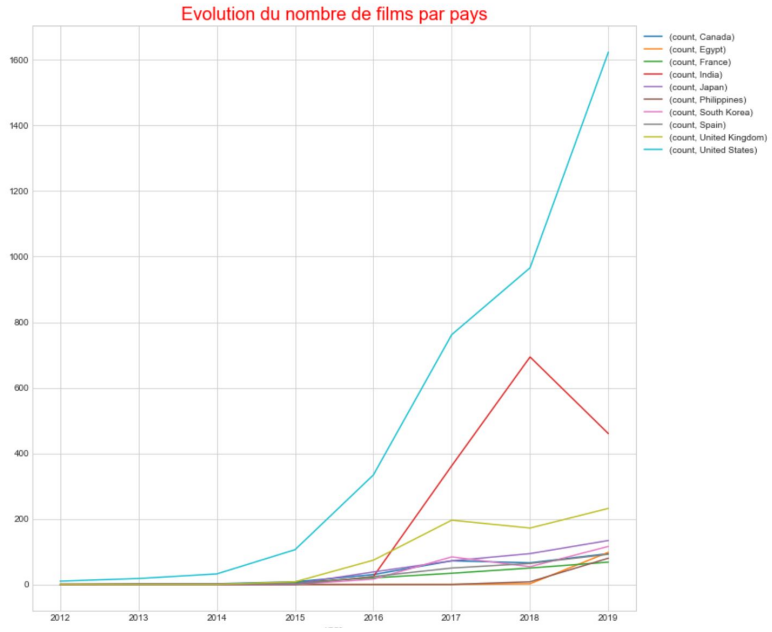
Voici le schéma relationnel des données.



Requêtes et analyse

Avec cette base de données, nous avons décidé d'analyser plusieurs questions.

Quels pays sont les plus gros producteurs de films ?

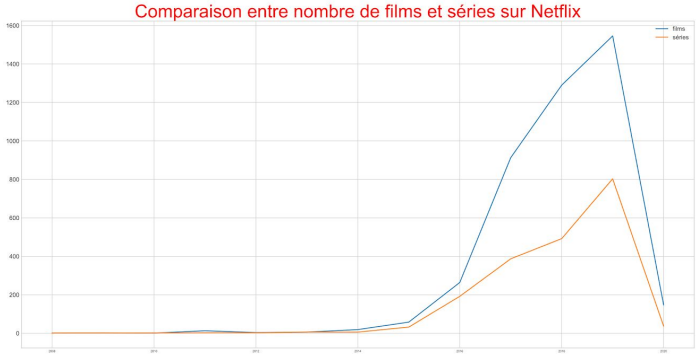


L'on remarque que les Etats-Unis sont le grand leader en terme de production de films, devant l'Inde.

```
request="""
MATCH (y:Year {value: 2012})-[:NEXT*0..10]->(y2:Year)<-[:CREATED_ON]-(f:Movie)-[r:WHERE]->(c:Country)
RETURN y2.value as year,c.name as country,count(r) as count
ORDER BY year DESC, count DESC
"""
```

Comment ont évolué les films face aux séries?

Nous avons décidé de visualiser le nombre de films et de séries ajoutés en fonction des années grâce à deux graphes superposés.

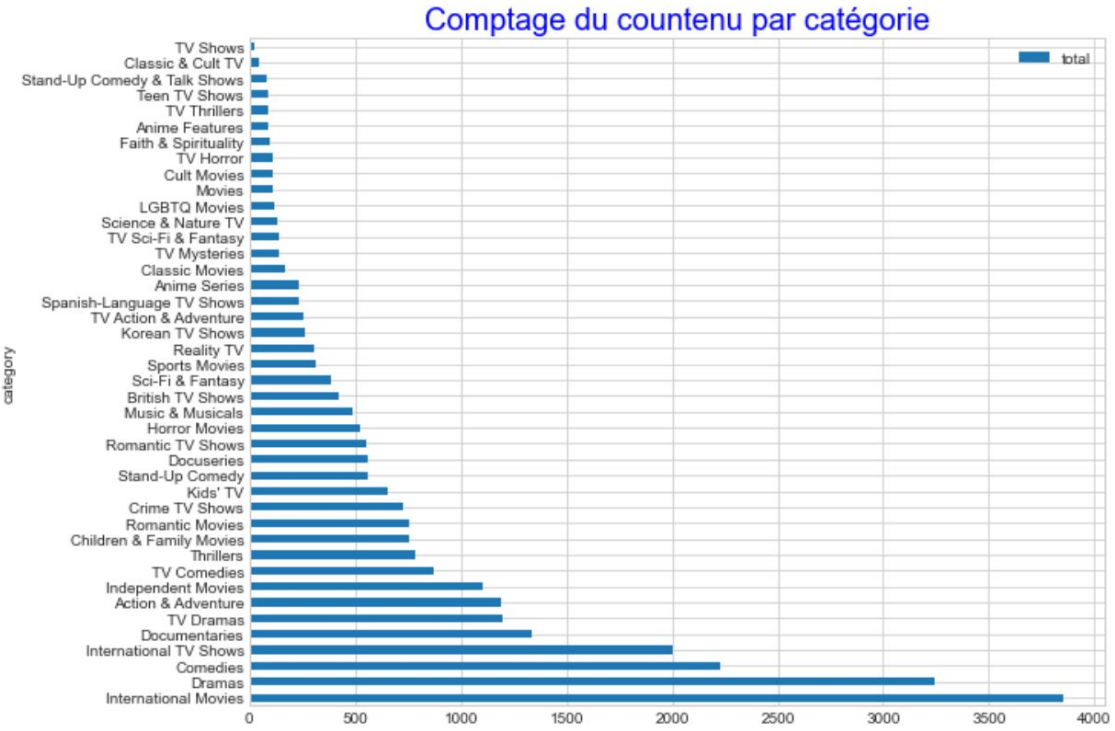


On peut voir que l'année 2019 a été l'année avec l'ajout du plus de films et de séries. 2018 semble avoir été plus propice pour les films que pour les séries.

```
request1="""MATCH (m:Movie)-[:CREATED_ON]->(y:Year)
WHERE m.type="Movie"
RETURN  y.value as year, count(m) as total
ORDER BY year
"""
```

```
request2="""MATCH (m:Movie)-[:CREATED_ON]->(y:Year)
WHERE m.type="TV Show"
RETURN  y.value as year2, count(m) as total2
ORDER BY year2
"""
```

Quelles sont les catégories prédominantes?



L'on remarque que les films internationaux sont loin devant ainsi que les dramas. Les comédies suivent de près.

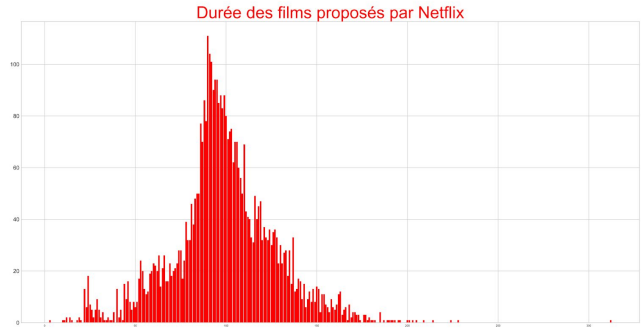
```
request="""
MATCH (c:Category)<-[rel:IN_CATEGORY]-(m:Movie)
WITH c.name as category,count(*) as total
RETURN  category,total
ORDER BY total DESC
"""
```

Quel est le rapport entre le nombre de films et l'année de sortie?

Netflix propose majoritairement des films de la dernière décennie, le plus ancien datant de 1941.

```
request3="""MATCH (m:Movie)-[:RELEASED_IN]->(y:date)
WITH y,count(*) as total3
RETURN  y.date as year3, total3
ORDER BY year3
"""
```

La durée des films présente-t-elle une particularité?



La durée des films semble être répartie similairement à une loi Normale centrée en environ 90 minutes avec une amplitude allant de 3 minutes à 5h12 avec l'épisode interactif de Black Mirror.

```
request4="""MATCH (m:Movie)-[:LASTS_IN_MINUTES]->(y:duration_film)
WHERE m.type="Movie"
WITH y,count(*) as total4
RETURN  y.duration_film as year4, total4
ORDER BY year4
"""
```

Qu’en est-il des séries?

Nous remarquons que les séries durent majoritairement une saison. Cependant, cela est dû au caractère récent de la plupart d’entre elles lorsqu’on regarde l’année de production des séries.

```
request5="""MATCH (m:Movie)-[:SEASONS]->(y:duration_serie)
WHERE m.type="TV Show"
WITH y,count(*) as total5
RETURN  y.duration_serie as year5, total5
ORDER BY year5
"""
```

```
request6="""MATCH (m:Movie)-[:RELEASED_IN]->(y:date)
WHERE m.type="TV Show"
WITH y,count(*) as total6
RETURN  y.date as year6, total6
ORDER BY year6
"""
```

Quels sont les acteurs ayant joué dans le plus de films?

```
$ MATCH (p:Person)-[rel:ACTED_IN]→(m:Movie) WITH p,collect(m.title) as movies,count(*) as total RE…
```

	p.name	movies	total
1	"Anupam Kher"	["Wake Up Sid", "A Wednesday", "Zokkomon", "C Kkompany", "Kyo Kii... Main Jhuth Nahin Bolta", "Kyaa Kool Hai Hum", "Aiyaary", "Chaahat", "Chashme Buddoor", "Special 26", "Yamla Pagla Deewana 2", "Hamara Dil Aapke Paas Hai", "Haseena Maan Jaayegi", "Judwaa 2", "Khalnayak", "Super Nani", "The Shaukeens", "Y.M.I.: Yeh Mera India", "Oh Darling Yeh Hai India", "Tahaan", "Hum Aapke Hain Koun", "Jaan-E-Mann: Let's Fall in Love... Again", "Judwaa", "The Indian Detective", "A Family Man", "Game", "Naam Shabana", "Toilet: Ek Prem Katha", "Khosla Ka Ghosla", "Rang De Basanti", "Mahabharat", "Kya Kehna", "Paheli"]	33
2	"Shah Rukh Khan"	["My Next Guest with David Letterman and Shah Rukh Khan", "Raees", "Zero", "Chamatkar", "Kabhi Haan Kabhi Naa", "Ram Jaane", "Chaahat", "English Babu Desi Mem", "Asoka", "One 2 Ka 4", "Dil Se", "Pardes", "Shakti: The Power", "Swades", "Trimurti", "Maya Memsaab", "Oh Darling Yeh Hai India", "Chalte Chalte", "Dilwale", "Happy New Year", "Jab Harry Met Sejal", "Don", "Don 2", "Chennai Express", "Dear Zindagi", "Billu", "Main Hoon Na", "Om Shanti Om", "Paheli", "Phir Bhi Dil Hai Hindustani"]	30
3	"Om Puri"	["Delhi 6", "Kismat Konnection", "Kurbaan", "Chaar Sahibzaade", "Chup Chup Ke", "Waarrior Savitri", "Action Replayy", "Oh My God", "Kyun! Ho Gaya Na", "Pitaah", "Pukar", "Sharaarat", "Yuva", "The Hundred-Foot Journey", "Mandi", "Mirzya", "Ghayal", "Road to Sangam", "Don", "Don 2", "Lakshya", "Viceroy's House", "Rang De Basanti", "Tere Naal Love Ho Gaya", "The Parole Officer", "Bollywood Calling", "Billu"]	27
4	"Naseeruddin Shah"	["Hope Aur Hum", "Main, Meri Patni Aur Woh", "SunGanges", "A Wednesday", "Aiyaary", "Chamatkar", "Kabhi Haan Kabhi Naa", "Michael", "Waiting", "Chaahat", "Dharam Sankat Mein", "Iqbal", "Mandi", "Peepli Live", "Shikari", "3 Deewarein", "Barah Aana", "Bazaar", "Encounter: The Killing", "John Day", "Katha", "Masoom", "Trikal (Past, Present, Future)", "Zindagi Na Milegi Dobara", "7 Khoon Maaf", "Raajneeti", "Main Hoon Na"]	27
5	"Yuki Kaji"	["DRAGON PILOT: Hisone & Masotan", "Berserk: The Golden Age Arc II - The Battle for Doldrey", "Berserk: The Golden Age Arc III - The Advent", "Berserk: The Golden Age Arc I - The Egg of the King", "BLAME!", "JoJo's Bizarre Adventure", "One-Punch Man", "B: The Beginning", "Your lie in April", "GODZILLA City on the Edge of Battle", "GODZILLA The Planet Eater", "Godzilla", "NiNoKuni", "Attack on Titan", "GANTZ:O", "Teasing Master Takagi-san", "The Seven Deadly Sins the Movie: Prisoners of the Sky", "The Disastrous Life of Saiki K.: Reawakened", "Fireworks", "Durarara!!", "Black Butler", "K", "Blue Exorcist", "The Seven Deadly Sins", "Haikyu!!", "Magi: The Labyrinth of Magic"]	26

```
requete_top_5="""
MATCH (p:Person)-[rel:ACTED_IN]->(m:Movie)
WITH p,collect(m.title) as movies,count(*) as total
RETURN  p.name, movies,total
ORDER BY total DESC
LIMIT 5
"""
```

On peut conjecturer que tous les acteurs retournés par la requête semblent être indiens, il aurait été intéressant d’avoir des informations supplémentaires les concernant (nationalité, tranche d’âge…) afin de vérifier cela.

Conclusion

Les données concernant Netflix sont nombreuses et diversifiées. Ainsi, la base de données Graphe est un très bon outil pour les représenter car elle permet de visualiser toutes les

relations existantes. Grâce au logiciel Neo4j, nous avons pu modéliser cette base de données et répondre à plusieurs questions précises.

Répartition des tâches

Travail commun sur le code et les requêtes par tout le groupe avec transmission de Neo4j aux autres membres par Exaucé et modélisation graphique majoritairement par Maxime et Zohra. Mise en page du rapport par Ekaterina et Merveille, création du diapo par Marie et Aida.