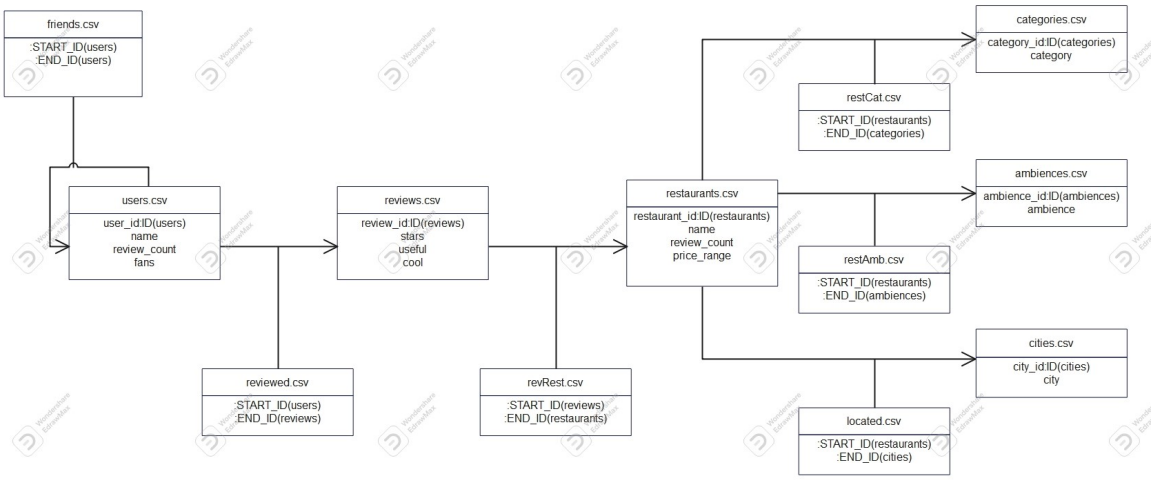
Compte rendu - Projet Neo4j

Enzo GREGOIRE - Quentin SIGNÉ

12 décembre 2022

1 Modélisation



1.1 Les noeuds

Notre modélisation comporte six noeuds qui sont users, reviews, restaurants, categories, ambiences et cities. Les tables suivantes décrivent plus en détail leurs champs.

Champs	Description
review_id	ID de la review.
stars	Note de la review en terme d'étoiles.
useful	Nombre de mentions "useful" obtenues par cette review.
cool	Nombre de mentions "cool" obtenues par cette review.

Table 1 – Reviews

Champs	Description
restaurant_id	ID du restaurant.
name	Nom du restaurant.
review_count	Nombre de reviews reçues par ce restaurant.
price_range	Catégorie de prix du restaurant.

Table 2 – Restaurants

Champs	Description
user_id	ID de l'utilisateur.
name	Nom de l'utilisateur.
review_count	Nombre de reviews effectuées par l'utilisateur.
fans	Nombre de fois où l'utilisateur a été "liké" sur la plateforme.

Table 3 – Users

Champs	Description
category_id	ID de la catégorie.
category	Nom de la catégorie.

Table 4 – Categories

Champs	Description
ambience_id	ID de l'ambiance.
ambience	Nom de l'ambiance.

Table 5 – Ambiences

Champs	Description
city_id	ID de la ville.
city	Nom de la ville.

Table 6 – Cities

1.2 Les relations

Notre modélisation comporte six relations :

Relationship	Start node	End node	Description
friends	users	users	Utilisateur et un de ses amis (pour tous ses amis).
reviewed	users	review	Utilisateur et une review qu'il a publié (pour toutes ces reviews).
revRest	reviews	restaurants	Review et restaurant associé.
restCat	restaurants	categories	Restaurant et une catégorie associée (pour toutes ces catégories).
restAmb	restraurants	ambiences	Restaurant et une ambiance associée (Pour toutes ces ambiances).
located	restaurants	cities	Restaurant et la ville associée.

Table 7 – Relationships

2 Interrogation des données

2.1 Donner le nombre de noeuds par label

```
MATCH (n:users) RETURN count(n)
```

Résultat : 23082 noeuds "users".

```
MATCH (n:cities) RETURN count(n)
```

Résultat : 18 noeuds "cities".

```
MATCH (n:reviews) RETURN count(n)
```

Résultat : 49150 noeuds "reviews".

```
MATCH (n:ambiences) RETURN count(n)
```

Résultat : 9 noeuds "ambiences"

```
MATCH (n:categories) RETURN count(n)
```

Résultat : 192 noeuds "categories"

```
MATCH (n:restaurants) RETURN count(n)
```

Résultat : 961 noeuds "restaurants"

Noeud	Résultat
users	23082
cities	18
reviews	49150
ambiences	9
categories	192
restaurants	961

TABLE 8 – Nombre de noeuds par label

2.2 Donner le nombre de relations par type

```
MATCH ()-[n:friends]->() RETURN count(n)
```

Résultat : 42776 relations "friends".

```
MATCH ()-[n:located]->() RETURN count(n)
```

Résultat : 961 relations "located".

```
MATCH ()-[n:restAmb]->() RETURN count(n)
```

Résultat : 505 relations "restAmb".

```
MATCH ()-[n:restCat]->() RETURN count(n)
```

Résultat : 4218 relations "restCat".

```
MATCH ()-[n:revRest]->() RETURN count(n)
```

Résultat : 49150 relations "revRest".

```
MATCH ()-[n:reviewed]->() RETURN count(n)
```

Résultat : 49150 relations "reviewed".

Relation	Résultat
friends	42776
located	961
restAmb	505
restCat	4218
revRest	49150
reviewed	49150

TABLE 9 – Nombre de relations par label

2.3 Donner la ou les catégories du restaurant *8th & Union Kitchen*

```
MATCH (res:restaurants)-[r:restCat]->(cat:categories)
WHERE res.name = "8th & Union Kitchen"
RETURN cat.category
```

Catégories du restaurant *8th & Union Kitchen* :
- gluten-free
- thai
- nightlife
- breakfast&brunch
- restaurants
- american(new)
- bars
- asianfusion
- vietnamese

2.4 Donner la ou les ambiances du restaurant *8th & Union Kitchen*

```
MATCH (res:restaurants)-[:restAmb]->(amb:ambiences)
WHERE res.name="8th & Union Kitchen"
RETURN amb.ambience
```

Ambiances du restaurant *8th & Union Kitchen* :
— casual
— trendy
— classy

2.5 Donner les reviews du restaurant *Tokyo Sushi*

```
MATCH (rev:reviews)-[r:revRest]->(res:restaurants)
WHERE res.name = "Tokyo Sushi"
RETURN rev
```

Résultats : Il y a 19 reviews pour le restaurant *Tokyo Sushi* (Figure 1).

FIGURE 1 – Graph des reviews du restaurant *Tokyo Sushi*

2.6 Donner les couples d'utilisateurs amis qui ont reviewé le restaurant *Tokyo Sushi*

```
MATCH (res:restaurants)<-[]-(:reviews)<-[]-(u1:users)-[:friends]->(u2:users)-[]->(:reviews)
-[]->(res:restaurants)
WHERE res.name="Tokyo Sushi"
RETURN DISTINCT u1.name,u2.name
```

Résultat : Seul le couple (Karen, Scott) (et (Scott, Karen)) sont des amis ayant reviewé le restaurant *Tokyo Sushi* (Figure 2).

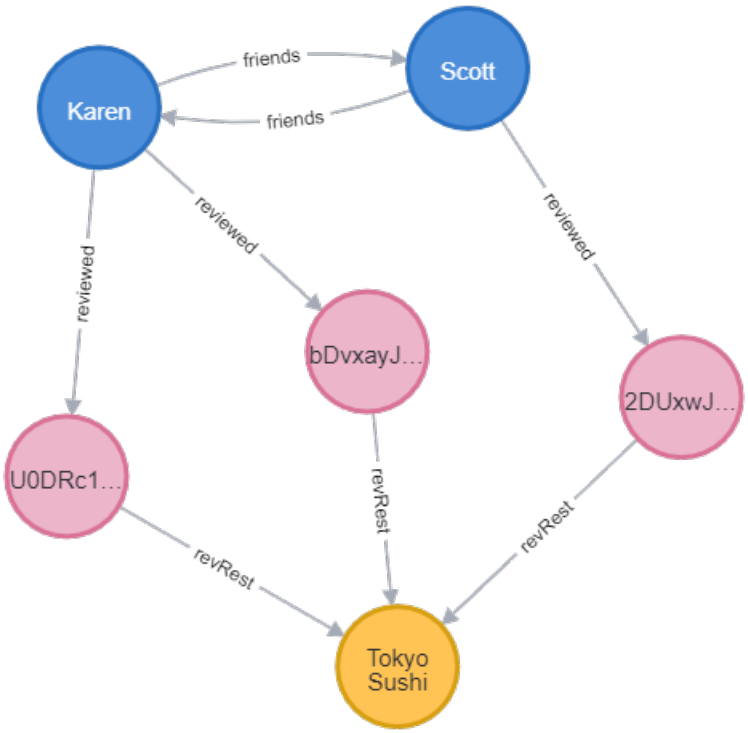


FIGURE 2 – Graph des couples d'amis qui ont reviewé le restaurant *Tokyo Sushi*

2.7 Qui est l'utilisateur qui a le plus d'amis ?

```
MATCH (u:users)-[r:friends]->(friend:users)
WITH u, count(friend) AS nb_friends
ORDER BY nb_friends DESC
LIMIT 1
RETURN u.name, nb_friends
```

Résultat : Morris est l'utilisateur avec le plus d'amis avec 461 amis.

2.8 Qui sont les amis des reviewers du restaurant Tokyo Sushi qui ont reviewés des restaurants avec les mêmes ambiances ?

```
MATCH (a:ambiences)<-[]-(r:restaurants)<-[]-(:reviews)<-[]-(u:users)-[:friends]->(v:users)
-[]->(:reviews)-[]->(x:restaurants)-[]->(a:ambiences)
WHERE r.name="Tokyo Sushi" and x <> r
RETURN DISTINCT v.name
```

Résultat : Il y a 221 utilisateurs ayant reviewé le restaurant *Tokyo Sushi*.

2.9 Combien de paires de personnes ont au moins 10 restaurants aimés en commun ?

```
MATCH (u1:users)-[]->(rev1:reviews WHERE toInteger(rev1.stars) >= 4)-[]->(res:restaurants)
<-[]-(rev2:reviews WHERE toInteger(rev2.stars) >= 4)<-[]-(u2:users)
WITH u1, u2, res, count(res) AS res_count
WHERE res_count >= 10
WITH count(u1) AS pair_count
RETURN pair_count
```

Résultat : 951 pairs ont au moins 10 restaurants aimés en commun.

3 Application : Moteur de recommandation d'influenceurs

Tous les tests sont réalisés avec $\alpha = 0.3$, $\beta = 0.3$, $\gamma = 0.3$ et $\delta = 0.1$.

Test n°	City	Ambiences	Categories	Price range
1	'Wilmington'	['casual']	['Pizza','Burgers','Italian']	1
2	'Wilmington'	['casual', 'romantic']	['Chinese']	2
3	'Wilmington'	['hipster']	['Nightlife','Bars']	1
4	'New Castle'	['casual','classy']	['Coffee & Tea']	2
5	'New Castle'	['classy']	['Seafood']	1

TABLE 10 – Paramètres des différents tests effectués

3.1 Test n°1

Temps d'exécution : 2.37 s

Rank	User_id	Name	Score
1	lHWT6Wv6B-GKvijrqn3zYQ	Raj	0.5500449038414542
2	nojn-VQzdVajcwXLiNzz5Q	marie	0.5333333333333332
3	Ug0rxREJ7nzndj3CnbWRUw	Sean	0.5275847993306018
4	wyObFWR0s6-3IhBQ5W6Gwg	Woody	0.5263785767746928
5	D5ek7jH87HEokWZ_K85dJQ	Namratha	0.5
5	3p-KctHIb2Oi4EobCWTAig	Reema	0.5
5	RLPZ1x9fADogVYzzhloWgg	Andrea	0.5
5	0Bp2fTDmIVK-FoXBntlJOw	Islam	0.5
5	QEyWj0Q_bMbChWVcwIVw5w	David	0.5
10	K5RgjPFhhtcuGD_SC3fY0g	Tim	0.4666666666666666

TABLE 11 – 10 meilleurs utilisateurs pour le test n°1

3.2 Test n°2

Temps d'exécution : 2.42 s

Rank	User_id	Name	Score
1	5zytUPH6qJ_Og-nO2lpyYA	Samantha	0.5499999999999999
2	7xiUQVDlRtarPTdwSCwReA	Dianne	0.5002345455438737
3	tRsqhChyRS5eqUoys40V0g	Tara	0.49345766010378234
4	t5JpuxLF9MlWlhebsl2yNA	Sonia	0.4672591383836551
5	_olgC2K6PYNYGOAFVsDMPw	Robert	0.4647459924713729
6	WnspMJ4Jd3nbpOFbHaKQfA	Sneha	0.46133684375289613
7	HQwOjVe11nPy-Q-q-20V-Q	John	0.4550914136411275
8	035JyP78qRqaiwY9EygoKQ	Laura	0.4547888129914168
9	YH80LMYyrmHUW-UVR3ncQQ	Annie	0.453947062149666
10	vvxgA0BPjiPwu1fsNEiTjQ	Jim	0.4503491132710261

TABLE 12 – 10 meilleurs utilisateurs pour le test n°2

3.3 Test n°3

Temps d'exécution : 2.42 s

Rank	User_id	Name	Score
1	tRsqhChyRS5eqUoys40V0g	Tara	0.4434576601037823
2	Ug0rxREJ7nzndj3CnbWRUw	Sean	0.4275847993306018
3	wyObFWR0s6-3IhBQ5W6Gwg	Woody	0.42637857677469276
4	MQntcxuGZ2eDWefjJGfV8A	Bain	0.4168604142803941
5	lHWT6Wv6B-GKvijrqn3zYQ	Raj	0.4167115705081209
6	_olgC2K6PYNYGOAFVsDMPw	Robert	0.41474599247137284
7	pyySLI0iM_YkeBuxrqZbKQ	Twig	0.4002345455438736
7	HJ-ql3WmYRb5gNSGSk8Ekw	Charlie	0.4002345455438736
9	ubRR1wg5CtbMimXbb5bYFA	Hana	0.40022573264178496
10	DpFM_a1HirNMCzirIr0LlQ	Jennifer	0.4000286286859433

Table 13 – 10 meilleurs utilisateurs pour le test n°3

3.4 Test n°4

Temps d'exécution : 2.36 s

Rank	User_id	Name	Score
1	KBTVrSLnxVQgiLpSMk2Z-w	Andrew	0.55
2	YH80LMYyrmHUW-UVR3ncQQ	Annie	0.5002433584459622
2	pyySLI0iM_YkeBuxrqZbKQ	Twig	0.5002345455438736
3	7xiUQVDlRtarPTdwSCwReA	Dianne	0.5002345455438736
3	KopKw_ATIlwMydkz3kfbUA	Jo	0.5
5	TFV3NvuoQzJKn1UbKfLsDw	Mia	0.5
7	tRsqhChyRS5eqUoys40V0g	Tara	0.4538350185943484
8	HQwOjVe11nPy-Q-q-20V-Q	John	0.4511266559318764
9	t5JpuxLF9MlWlhebsl2yNA	Sonia	0.45059247171698846
10	_olgC2K6PYNYGOAFVsDMPw	Robert	0.4504602781856586

Table 14 – 10 meilleurs utilisateurs pour le test n°4

3.5 Test n°5

Temps d'exécution : 2.2 s

Rank	User_id	Name	Score
1	D5ek7jH87HEokWZ_K85dJQ	Namratha	0.5
1	TFV3NvuoQzJKn1UbKfLsDw	Mia	0.5
3	lHWT6Wv6B-GKvijrqn3zYQ	Raj	0.40286541666196707
4	MQntcxuGZ2eDWefjJGfV8A	Bain	0.40271900013898
5	Ug0rxREJ7nzndj3CnbWRUw	Sean	0.40258479933060176
6	wyObFWR0s6-3IhBQ5W6Gwg	Woody	0.40176319215930817
7	YH80LMYyrmHUW-UVR3ncQQ	Annie	0.40024335844596226
8	7xiUQVDlRtarPTdwSCwReA	Dianne	0.4002345455438736
9	ubRR1wg5CtbMimXbb5bYFA	Hana	0.40022573264178496
10	DpFM_a1HirNMCzirIr0LlQ	Jennifer	0.4000286286859433

Table 15 – 10 meilleurs utilisateurs pour le test n°5