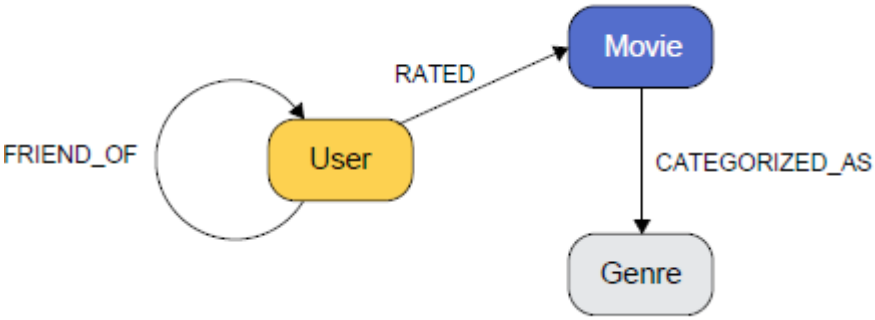
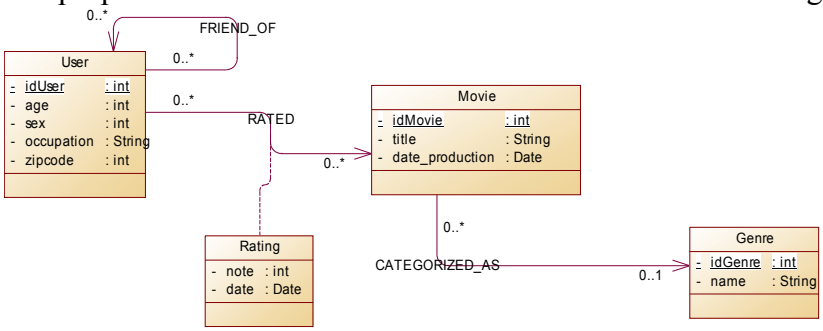
Exercice (disponible sur les ressources de neo4j) :

Nous souhaitons construire la base de données disponible dans les fichiers CSV fournis. Cette base de données suit le schéma graphe suivant :



Les propriétés des nœuds et des arêtes sont décrites dans le diagramme UML suivant :



Mise en place de la base :

- Copier le répertoire « import » dans le répertoire « database location »

L'instruction qui permet d'importer un fichier csv est (http://neo4j.com/docs/stable/cypherdoc-importing-csv-files-with-cypher.html):

```
LOAD CSV WITH HEADERS FROM 'file:///users.csv' AS users
CREATE (u:User {id : toInt(users.id), sex : users.sex, age : toInt(users.age), occupation : users.occupation, zip_code : users.zip_code});
```

1- Importer Movies et Genres

2- Pour importer les autres fichiers, il faut noter que ces fichiers correspondent à des associations. Par conséquent, on crée des arcs. De plus, les nœuds existent, il ne faut pas les recréer : utiliser l'option merge

```
USING PERIODIC COMMIT
LOAD CSV WITH HEADERS FROM 'file:D:\\neo4j\\TP\\mov_genre.csv' AS mov_genre
MERGE (m : Movie { id : toInt(mov_genre.mov_id) })
MERGE (g : Genre { id : toInt(mov_genre.genre) })
CREATE (m)-[:CATEGORIZED_AS]->(g);
```

En s'aidant de la documentation sur CYPHER (https://neo4j.com/docs/developer-manual/current/cypher/) et les différentes options de la commande MATCH (http://neo4j.com/docs/stable/query-match.html#_basic_node_finding), donnez les requêtes permettant de :

Requêtes simples

1. Lister les 200 premières données de la base (nœuds et leur relations)
2. Lister tous les users
3. Donner L'âge de l'utilisateur dont l'id est 5
4. Afficher les 20 premiers films
5. Afficher les films qui ont été notés par l'utilisateur dont l'id est 1
6. Afficher les films qui ont été notés par l'utilisateur dont l'id est 1 ainsi que leur genres (limiter l'affichage à 10)
7. Afficher les notes des films de genre « Drama »
8. Lister les utilisateurs de sexe féminin qui sont soit écrivaines soit artistes
9. Quel est l'âge moyen des étudiants ? (voir la notion d'agrégation https://neo4j.com/docs/developer-manual/current/cypher/functions/aggregating/)
10. Quel est l'âge moyen par occupation ?
11. Quelles sont les 3 « occupations » les plus populaires ? https://neo4j.com/docs/developer-manual/current/cypher/clauses/order-by/
12. Combien de valeurs différentes existent-ils pour l'attribut occupation ?
13. Quels sont les films produits en 1995 ? (la date de production est contenue dans le titre du film)
14. Lister les associations avec leur nombres d'occurrences (TYPE(r) retourne le type d'une association r).
15. Combien de notations y-a-t-il (i.e., combien d'occurrences de l'association RATED)
16. Quels sont les 5 films les plus notés
17. Combien de films ont reçu au moins une fois la note 1
18. Donner la liste des films qui ont reçu au moins une fois la note 1 en donnant le nombre de fois où ils ont reçu cette note et qui a donné la note (utiliser la fonction agrégative ***collect*** https://neo4j.com/docs/developer-manual/current/cypher/functions/aggregating/#functions-collect).
19. Quels sont les films qui ont une note moyenne >4

(Pour filtrer un agrégat, vous pouvez utiliser la clause ***WITH*** http://neo4j.com/docs/developer-manual/current/cypher/clauses/with/)

20. Quels sont les films regardés par le user 13
21. Quels sont les films non regardés par le user 13 ? (les comparaisons et négations sont similaires au SQL, utiliser NOT)
22. Lister le nombre d'associations par type de nœud (labels(x) retourne le type du nœud x)

Requêtes sur les chemins :

23. Lister les amis et les amis des amis du user 1 (Attention user 1 ne doit pas être ami de lui-même)
24. (vous pouvez utiliser un pattern de chemin de longueur 2 : voir cours ou http://neo4j.com/docs/developer-manual/current/cypher/syntax/patterns/)

25. Lister les amis et les amis des amis du user 1 en donnant leur distance du user 1 (cette distance est 1 lorsque c'est un ami direct et est 2 lorsque c'est un ami d'un ami).
26. Vous pouvez utiliser une Union https://neo4j.com/blog/cypher-union-query-using-collect-clause/

27. Quel est l'utilisateur qui a le plus d'amis en communs avec l'utilisateur 41.

28. Quel utilisateur a noté le plus grand nombre de films que l'utilisateur 1 a également notés et quel est ce nombre ?