Alexander Powers
Prof Guadalupe Canahuate
ECE:5995 Modern Databases
November 19th 2020

# Homework Six

## Q1.0

```
:auto USING PERIODIC COMMIT 500
LOAD    CSV WITH HEADERS FROM "file:///ml-latest-small/movies.csv"  AS row
MERGE   (m:Movie  {movieId: toInteger(row.movieId), title: row.title})
WITH    m, row
UNWIND  split(row.genres, '|')  AS genres
MERGE   (g:Genre {name: genres})
MERGE   (m)-[r:IN_GENRE]->(g);
```

## Q1.1

```
CREATE  INDEX MovieIdIndex
FOR     (n:Movie)
ON      (n.movieid)
```

## Q1.2

```
:auto USING PERIODIC COMMIT 500
LOAD    CSV WITH HEADERS FROM "file:///ml-latest-small/ratings.csv"  AS row
MERGE   (u:User  {userId: toInteger(row.userId)})
WITH    u, row
MATCH   (m:Movie {movieId: toInteger(row.movieId)})
MERGE   (u)-[r:RATED {rating: toInteger(row.rating), timestamp: row.timestamp} ]->(m)
```

## Q2.1

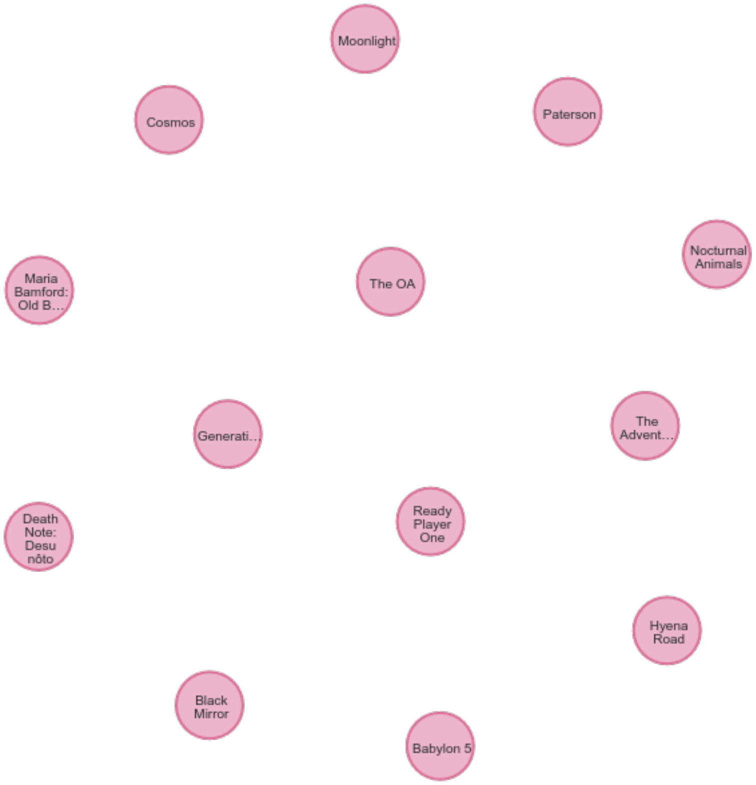
```
MATCH   (m:Movie)
WHERE   m.title =~ '.*\(\d{4}\).*'
SET     m.year = toInteger(substring(apoc.text.replace(m.title, '[^\d]', ''),
size(apoc.text.replace(m.title, '[^\d]', ''))-4, 4))
RETURN  COUNT(m)
```

## Q2.2

```
MATCH   (m:Movie)
WHERE   m.year IS NULL
RETURN  m
```

```
MATCH   (m:Movie)
WHERE   m.year IS NULL
```

```
RETURN  count(m)
```



## Q2.3

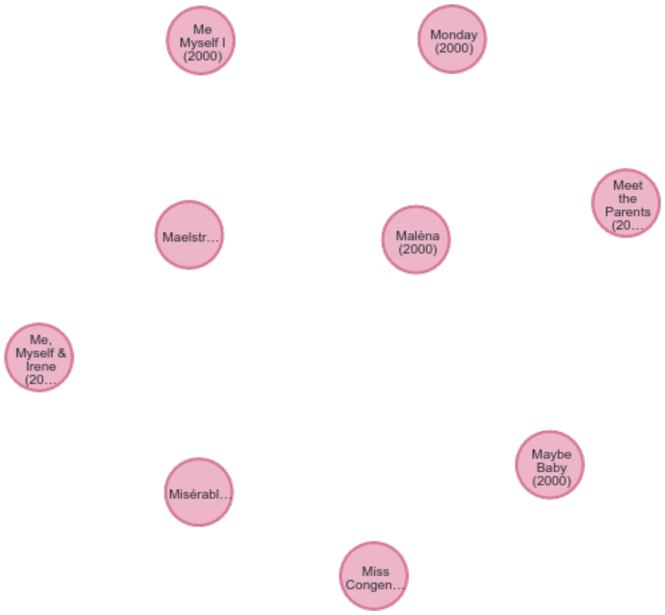
```
MATCH   (m:Movie)
WITH    MIN(m.year) as old, MAX(m.year) as new
UNWIND  range(old, new, 10) as startDecade
WITH    startDecade, startDecade + 9 as endDecade
MATCH   (movie:Movie)
WHERE   movie.year >= startDecade and movie.year <= endDecade
RETURN  startDecade + "-" + endDecade as years, count(movie)
ORDER   BY years
```

```
years,count(movie)
1902-1911,3
1912-1921,10
1922-1931,53
1932-1941,160
1942-1951,197
1952-1961,307
1962-1971,410
1972-1981,601
1982-1991,1290
1992-2001,2495
```

```
2002-2011,2773
2012-2021,1430
```

## Q2.4

```
MATCH   (m:Movie)
WHERE   m.year = 2000 AND (m.title STARTS WITH 'M') AND ((m)-[:IN_GENRE]-(:Genre
{name: 'Comedy'}) OR (m)-[:IN_GENRE]-(:Genre {name:'Romance'}))
RETURN  m;
```



## Q3.1

```
MATCH   (m:Movie)
WHERE   NOT (m)-[:IN_GENRE]->(:Genre)
RETURN  m

MATCH   (m:Movie)
WHERE   NOT (m)-[:IN_GENRE]->(:Genre)
RETURN  COUNT(m)
```

## Q3.2

```
MATCH   (m:Movie)-[:IN_GENRE]->(g:Genre)
RETURN  g.name, count(m) as movieCount
ORDER   BY movieCount DESC
```

```
g.name,movieCount
Drama,4361
Comedy,3756
Thriller,1894
Action,1828
Romance,1596
```

```
Adventure,1263
Crime,1199
Sci-Fi,980
Horror,978
Fantasy,779
Children,664
Animation,611
Mystery,573
Documentary,440
War,382
Musical,334
Western,167
IMAX,158
Film-Noir,87
(no genres listed),34
```

## Q3.3

```
MATCH       (m:Movie)<-[r:RATED]-(u:User)
RETURN      m.title, COUNT(r) as numReviews
ORDER BY    numReviews DESC
LIMIT       5
```

```
m.title,numReviews
Forrest Gump (1994),329
"Shawshank Redemption, The (1994)",317
Pulp Fiction (1994),307
"Silence of the Lambs, The (1991)",279
"Matrix, The (1999)",278
```

## Q3.4

```
MATCH       (g:Genre)<-[:IN_GENRE]-(m:Movie)<-[r:RATED]-(u:User)
RETURN      m.title, AVG(r.rating) as movieRating, collect(g.name)
ORDER BY    movieRating DESC
LIMIT       10
```

```
m.title,movieRating,collect(g.name)
Ice Age: The Great Egg-Scapade (2016),5.0,"[Adventure,Animation,Children,Comedy]"
The Love Bug (1997),5.0,"[Adventure,Children,Comedy,Fantasy]"
12 Chairs (1976),5.0,"[Adventure,Comedy]"
Junior and Karlson (1968),5.0,"[Adventure,Animation,Children]"
On the Trail of the Bremen Town Musicians (1973),5.0,"[Adventure,Animation,Children]"
Priklyucheniya Kapitana Vrungelya (1979),5.0,"[Adventure,Animation,Comedy,Action]"
Karlson Returns (1970),5.0,"[Adventure,Animation,Children]"
Dragons: Gift of the Night Fury (2011),5.0,"[Adventure,Animation,Comedy]"
```

```
Vovka in the Kingdom of Far Far Away (1965),5.0,"[Adventure,Animation,Children,Fantasy]"
The Fox and the Hound 2 (2006),5.0,"[Adventure,Animation,Children,Comedy]"
```

## Q3.5

```
MATCH   (m:Movie)<-[r:RATED]-(u:User {userId: 3})
RETURN  COLLECT(m.title)
```

```
COLLECT(m.title)
"[Bambi (1942),Rescuers, The (1977),Death Race 2000 (1975),Android (1982),Patton
(1970),Requiem for a Dream (2000),Fast Times at Ridgemont High (1982),Piranha
(1978),Doors, The (1991),Clonus Horror, The (1979),Troll 2 (1990),Road Warrior, The (Mad
Max 2) (1981),The Lair of the White Worm (1988),Thing, The (1982),Escape from L.A.
(1996),Deer Hunter, The (1978),Looker (1981),Field of Dreams (1989),Highlander (1986),Star
Trek: The Motion Picture (1979),2012 (2009),Alien Contamination (1980),Galaxy of Terror
(Quest) (1981),Conan the Barbarian (1982),Courage Under Fire (1996),My Fair Lady
(1964),Lady and the Tramp (1955),Green Card (1990),On Golden Pond (1981),Wallace &
Gromit: The Best of Aardman Animation (1996),Schindler's List (1993),Saturn 3 (1980),You've
Got Mail (1998),Dangerous Minds (1995),Operation Dumbo Drop (1995),Tron (1982),Snow
Dogs (2002),Hangar 18 (1980),Master of the Flying Guillotine (Du bi quan wang da po xue di
zi) (1975)]"
```

## Q3.6

```
MATCH   (u:User {userId:1})-[r:RATED]->(m:Movie)-[:IN_GENRE]->(g:Genre)
RETURN  g.name, AVG(r.rating)
```

```
g.name,AVG(r.rating)
Western,4.285714285714286
Comedy,4.277108433734937
Adventure,4.388235294117646
Sci-Fi,4.225000000000001
Action,4.322222222222222
War,4.500000000000001
Drama,4.529411764705883
Thriller,4.1454545454545455
Musical,4.681818181818182 5
Fantasy,4.29787234042553
Horror,3.4705882352941178
Children,4.547619047619048
Mystery,4.166666666666667
Crime,4.355555555555555
Film-Noir,5.0
Animation,4.689655172413794
Romance,4.3076923076923075
```

## Q4.1

```
MATCH (u1:User)-[x:RATED]->(m:Movie)<-[y:RATED]-(u2:User)
WITH  SUM(x.rating * y.rating) AS xyDotProduct,
     SQRT(REDUCE(xDot = 0.0, a IN COLLECT(x.rating) | xDot + a^2)) AS xLength,
     SQRT(REDUCE(yDot = 0.0, b IN COLLECT(y.rating) | yDot + b^2)) AS yLength,
     u1, u2
MERGE (u1)-[s:SIMILARITY]-(u2)
SET   s.similarity = xyDotProduct / (xLength * yLength)
```

## Q4.2

```
MATCH      (u:User {userId:1})-[s:SIMILARITY]-(u2:User)
RETURN     u2.userId, s.similarity
ORDER BY   s.similarity DESC
LIMIT      10
```

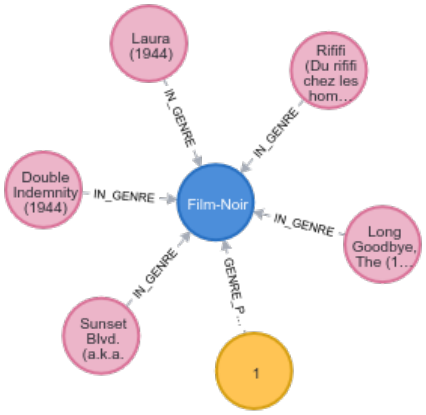
```
u2.userId,s.similarity
85,1.0
388,1.0
383,1.0
259,1.0
184,1.0
315,1.0
358,1.0
12,1.0
511,1.0
77,1.0
```

## Q4.3

```
MATCH   (u:User)-[r:RATED]->(:Movie)-[:IN_GENRE]->(g:Genre)
WITH    u, g, AVG(r.rating) AS avgRating
MERGE   (u)-[p:GENRE_PREF]->(g)
SET     p.preference = avgRating
```

## Q5.1

```
MATCH   (u:User {userId: 1})-[gp:GENRE_PREF]->(g:Genre)<-[ig:IN_GENRE]-(m:Movie)<-
[r:RATED]-(u2:User)
WHERE   NOT((u)-[:RATED]->(m))
RETURN  u, max(gp.preference) AS pref, g, AVG(r.rating) as movieRating, m
ORDER   BY pref DESC, movieRating DESC
LIMIT   5
```



## Q5.2

```
MATCH    (b:User)-[r:RATED]->(m:Movie), (b)-[s:SIMILARITY]-(a:User {userId:0})
WHERE    NOT((a)-[:RATED]->(m))
WITH     m, s.similarity AS similarity, r.rating AS rating
ORDER BY m.name, similarity DESC
WITH     m.name AS movie, COLLECT(rating)[0..3] AS ratings
WITH     movie, REDUCE(s = 0, i IN ratings | s + i)*1.0 / SIZE(ratings) AS reco
ORDER BY reco DESC
LIMIT    10
RETURN   movie AS Movie, reco AS Recommendation
```