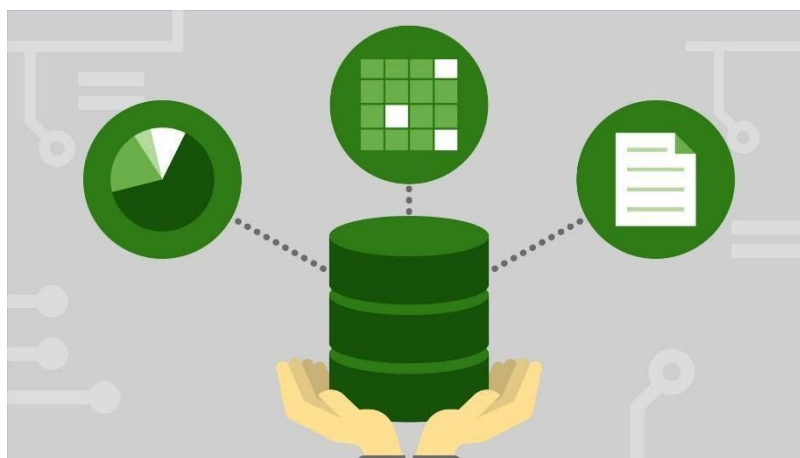


به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



آزمایشگاه پایگاه داده
دستور کار شماره ۳

شماره دانشجویی

۸۱۰۱۹۶۶۰۴

آبان ۹۹

نام و نام خانوادگی
سید پارسا حسینی نژاد

گزارش فعالیت‌های انجام شده

برای انجام پروژه، ابتدا لازم است که دیتابیس Neo4j را نصب کنیم. پس از نصب با استفاده از این [لینک](#)، داده‌ها را در دیتابیس لود می‌کنیم. برای این کار، دیتاست MovieLens-small را دانلود کرده و کد پایتون مربوطه را ران می‌کنیم تا دیتا در دیتابیس پر شود. این پروسه‌ی لود شدن حدود ۲ ساعت به طول انجامید.

حال درخواست‌های ذکر شده را جواب می‌دهیم:

1. ژانرهای دیتاست:

`Collect()` لیستی را که حاوی مقادیر برگشتی توسط یک عبارت است، برمی‌گرداند. با استفاده از این تابع داده‌ها را با ادغام چندین رکورد با مقدار، در یک لیست واحد جمع می‌کند. پس در واقع با این تابع، تمامی ژانرهای مختلف را استخراج می‌کنیم. در واقع در این کوئری، در بین تمام ژانرهای g، نام آن‌ها را برمی‌گردانیم.

```
MATCH (g:Genre) WITH collect(g.name) as genres RETURN genres
```

genres

["Thriller", "War", "Western", "Action", "Adventure", "Animation", "Children's", "Comedy", "Crime", "Documentary", "Drama", "Fantasy", "Film-Noir", "Horror", "Musical", "Mystery", "Romance", "Sci-Fi"]

2. تعداد کل فیلم‌ها:

در این کوئری، با استفاده از تابع `count()`، تعداد تمامی فیلم‌ها را می‌یابیم.

```
neo4j$ MATCH (:Movie) WITH count(*) as count RETURN count
```

	count
1	9742

3. تعداد امتیازات داده شده به The silence of the lambs در هر رده:

در این کوئری، $MATCH (:User)-[r:RATED] \rightarrow (m:Movie)$ به این معناست که داده ها کاربران و فیلم ها هستند و relation بین آن ها همان Rated می باشد. حال پس از پیدا کردن تعداد هر دسته، آن را بر اساس rating مرتب می کنیم.

```
j$ MATCH (:User)-[r:RATED]→(m:Movie) WHERE m.title="Silence of the Lambs, The" WITH r.rating as rating, count(*) as count RETURN rating, count
```

	rating	count
1	0.5	3
2	1.0	1
3	2.0	5
4	2.5	7
5	3.0	24
6	3.5	14
7	4.0	97
8	4.5	36
9	5.0	92

tarted streaming 9 records after 1 ms and completed after 84 ms.

4. تعداد امتیازات هر فیلم به صورت نزولی:

این کوئری هم مانند قبلی است که relation دو نود مربوط به Rated می باشد.

```
j$ MATCH (:User)-[r:RATED]→(m:Movie) WITH m.id as id, m.title as title, count(*) as count RETURN title, count ORDER BY count DESC
```

	title	count
1	"Forrest Gump"	329
2	"Shawshank Redemption, The"	317
3	"Pulp Fiction"	307
4	"Silence of the Lambs, The"	279
5	"Matrix, The"	278
6	"Star Wars: Episode IV - A New Hope"	251

rted streaming 9724 records after 1 ms and completed after 2 ms, displaying first 1000 rows.

5. سه بخش زیر:

1. ژانر با بیشترین فیلم:

در این قسمت رابطه‌ی بین فیلم‌ها و ژانر ها که از جنس genre هستند را در نظر گرفته، آن را بر اساس نزولی مرتب کرده و بزرگترین عدد را برمی‌گردانیم.

```
neo4j$ MATCH (g:Genre)-[:IS_GENRE_OF]-(m:Movie) WITH g.name as genre, count(*) as count ORDER BY count DESC LIMIT 1 RETURN genre
```

genre
1 "Drama"

2. ژانر با بیشترین امتیاز:

در این کوئری، هم رابطه‌ی genre و هم رابطه‌ی Rated را در نظر گرفته تا ژانری که میانگین امتیاز بیشینه دارد را به دست آوریم. از تابع avg() هم برای میانگین گیری استفاده می‌کنیم.

```
neo4j$ MATCH (g:Genre)-[:IS_GENRE_OF]-(m:Movie)-[r:RATED]-(u:User) WITH g.name as genre, avg(r.rating) as avgrating ORDER BY avgrating DESC LIMIT 1 ...
```

genre
1 "Film-Noir"

3. ژانر با کمترین امتیاز:

مانند کوئری قبل است و صرفاً ترتیب مرتب سازی فرق دارد.

```
neo4j$ MATCH (g:Genre)-[:IS_GENRE_OF]-(m:Movie)-[r:RATED]-(u:User) WITH g.name as genre, avg(r.rating) as avgrating ORDER BY avgrating LIMIT 1 RETURN...
```

genre
1 "Horror"

6. فیلم‌های تولید شده در سال 2000:

برای این کوئری کفیفست در نودهای فیلم آنهایی که سال تولید 2000 دارند را برگردانیم.

```
neo4j$ MATCH (m:Movie) WHERE m.year="2000" WITH m.title as title RETURN title
```

	title
1	"Yards, The"
2	"Next Friday"
3	"Supernova"
4	"Down to You"
5	"Scream 3"
6	"Boondock Saints, The"
7	

Started streaming 283 records in less than 1 ms and completed after 6 ms.

7. فیلم با کمترین امتیاز:

از relation مربوط به Rated استفاده می‌کنیم و میانگین امتیاز هر فیلم را محاسبه می‌کنیم.

```
j$ MATCH (u:User)-[r:RATED]-(m:Movie) WITH m.id as id, m.title as title, count(*) as count, avg(r.rating) as avgrating RETURN title, avgrating ...
```

	title	avgrating
1	"Rust and Bone (De rouille et d'os)"	0.5

8. تعداد کل کاربران:

```
j$ MATCH (:User) WITH count(*) as count RETURN count
```

	count
1	610

9. بیست فیلم محبوب با این شرط که حداقل 30 نفر به آن رای داده باشند:

برای این کوئری شرط حداقل 30 نفر را اعمال کرده و بر اساس میانگین امتیاز مرتب کرده و 20 تای اول را برمیگردانیم.

```
neo4j$ MATCH (u:User)-[r:RATED]->(m:Movie) WITH m.id as id, m.title as title, count(*) as count, avg(r.rating) as avgrating WHERE count ≥ 100 RETURN ...
```

	title
1	"Shawshank Redemption, The"
2	"Godfather, The"
3	"Fight Club"
4	"Godfather: Part II, The"
5	"Departed, The"
6	"Goodfellas"
7	...

Started streaming 20 records after 1 ms and completed after 216 ms.