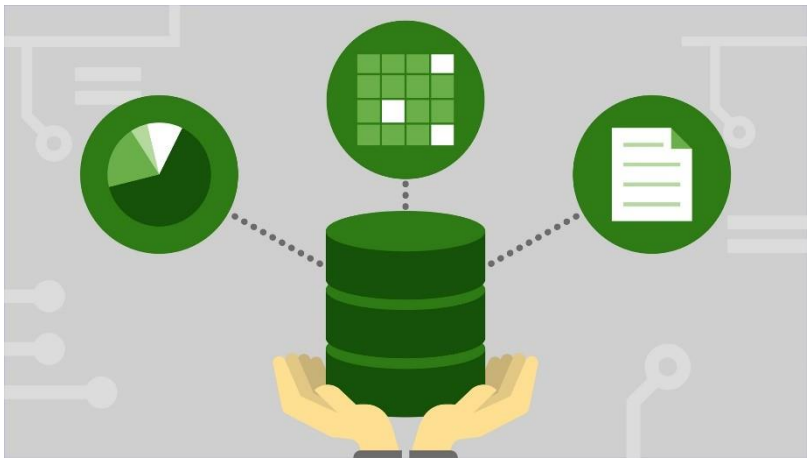
به نام خدا

دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



آزمایشگاه پایگاه داده

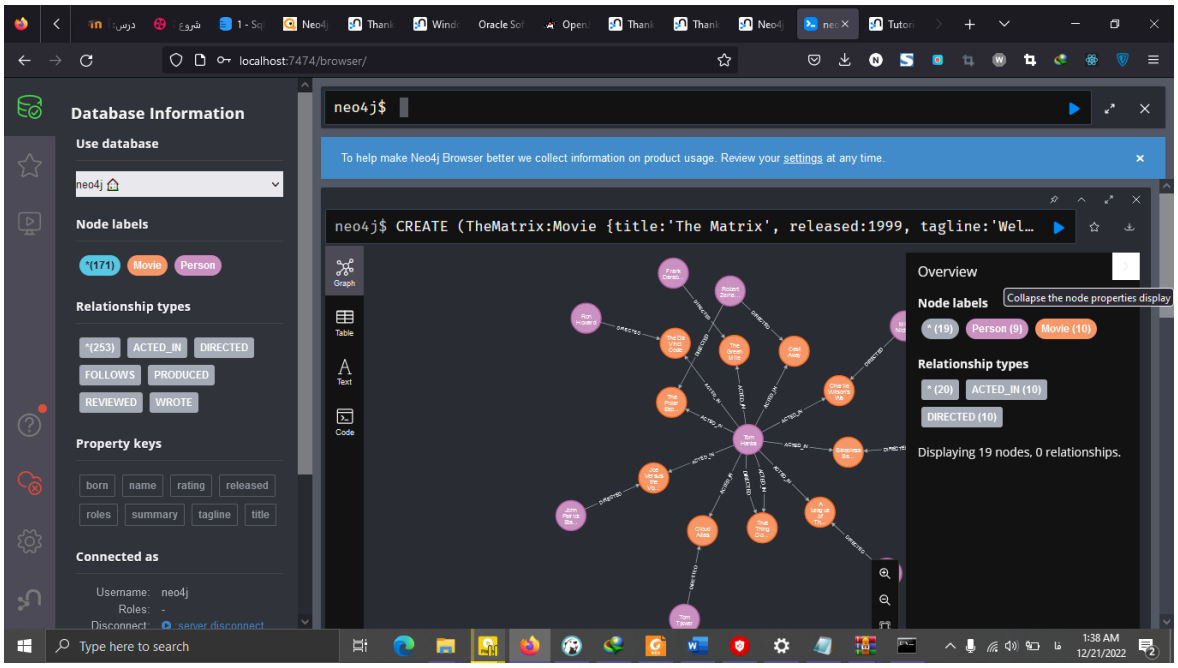
دستور کار شماره 3

نگار مرادی
810198543

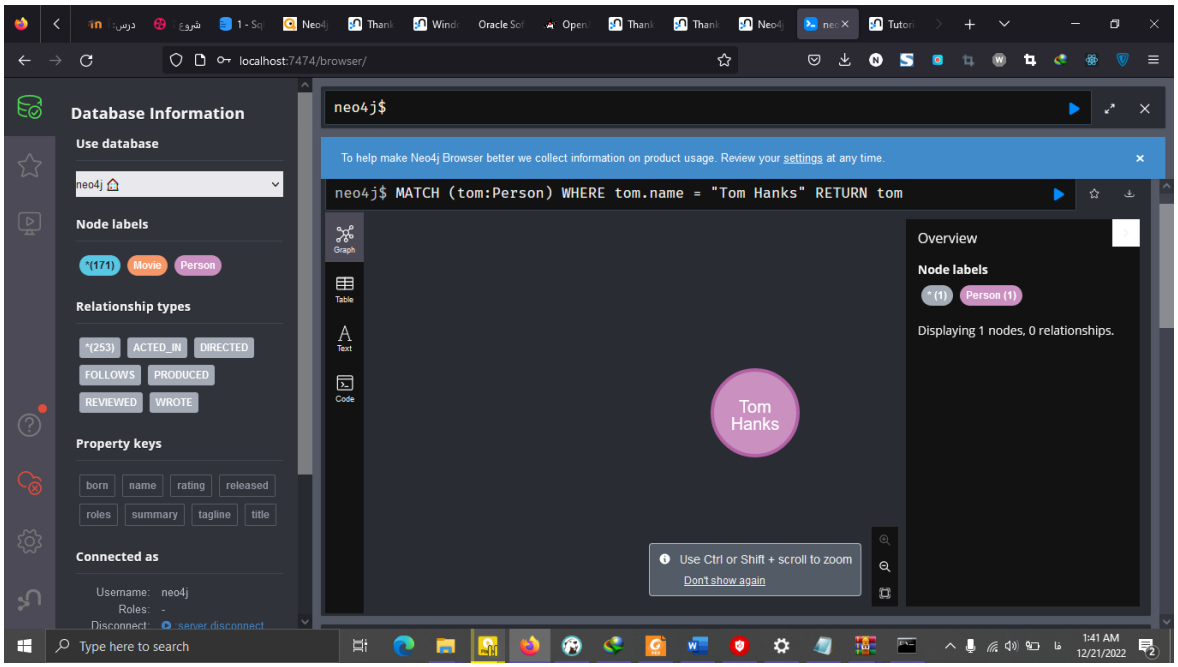
مهر 1401

گزارش دستورکار انجام شده

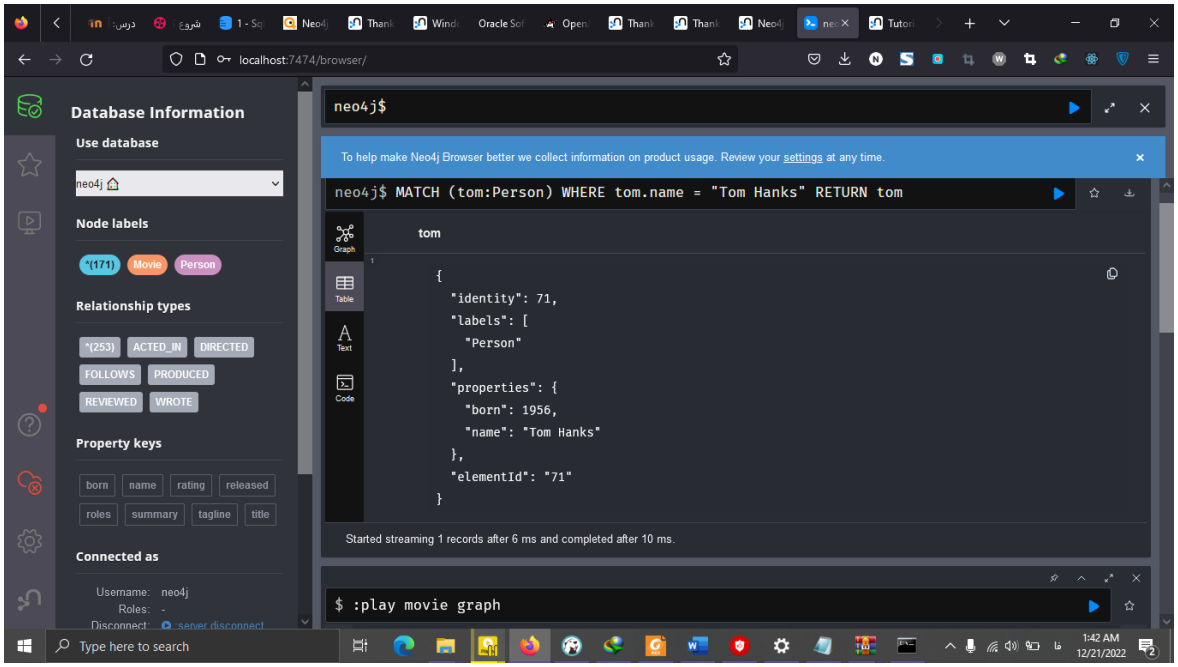
طبق دستورات دیتابیس را لود کردیم



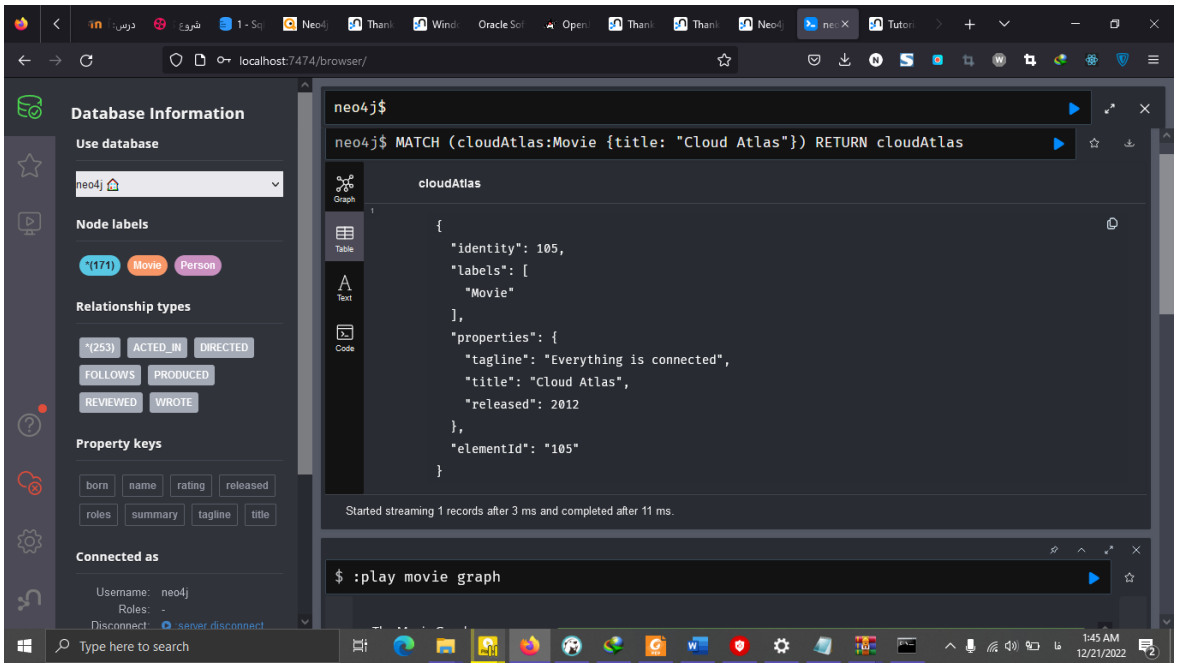
دستور بعدی ادمی با نام tom hanks را پیدا میکند. در این دستور تمام person هایی که نام آن ها tom hanks هست را پیدا میکند. و در انتها یک نود person که name آن تام هنکس است را باز میگرداند.



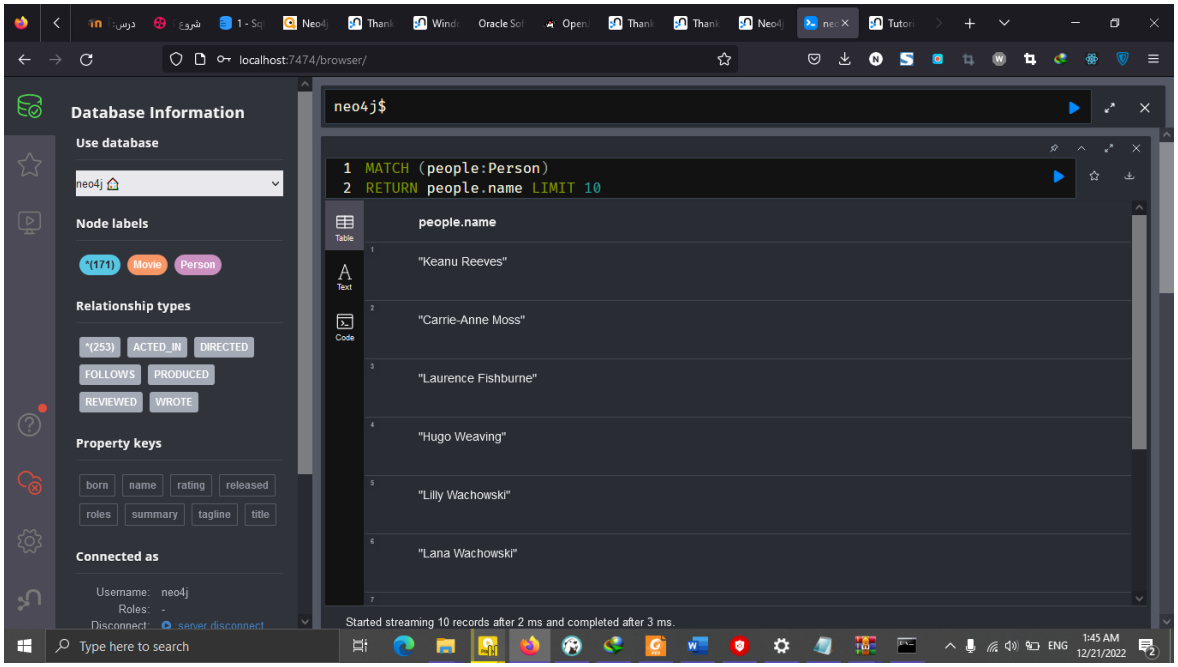
همچین جدول آن را می توانیم مشاهده کنیم:



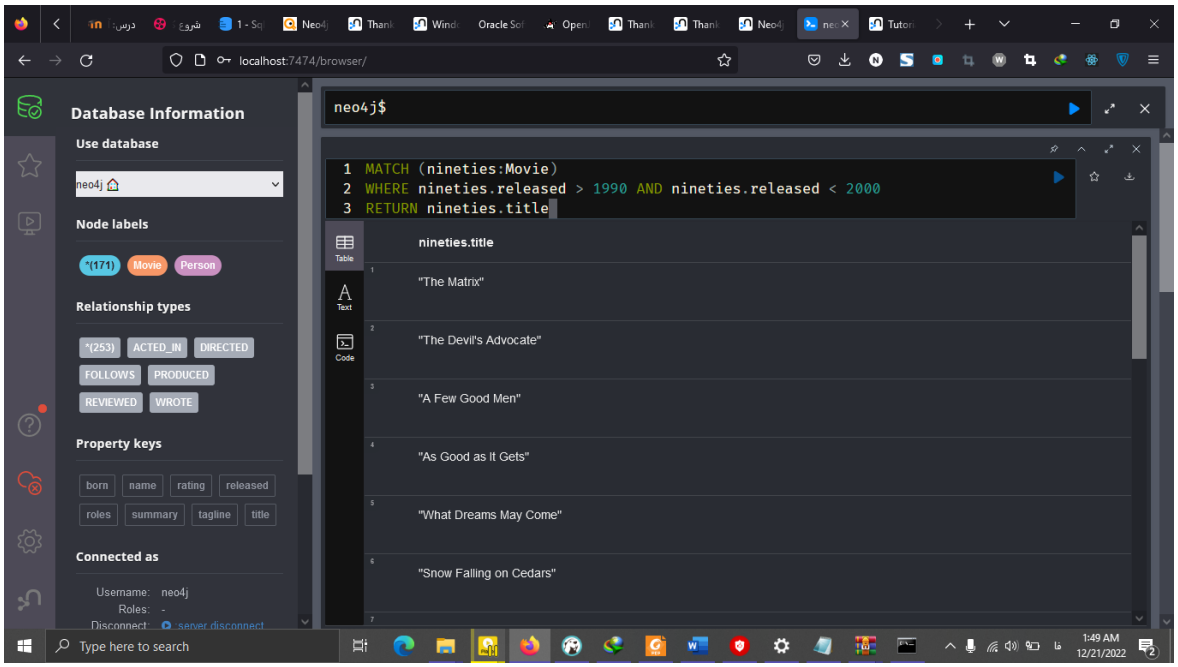
پیدا کردن فیلم با نام Cloud Atlas مانند مثال قبل تمام نودهای movie که title آن ها برابر cloud atlas است را پیدا میکنیم و آن نود را بازمیگردانیم.



پیدا کردن اسم 10 نفر تا اندازه 10 نفر return person میکنیم. 10 خروجی اول تابع match را return میکنیم.

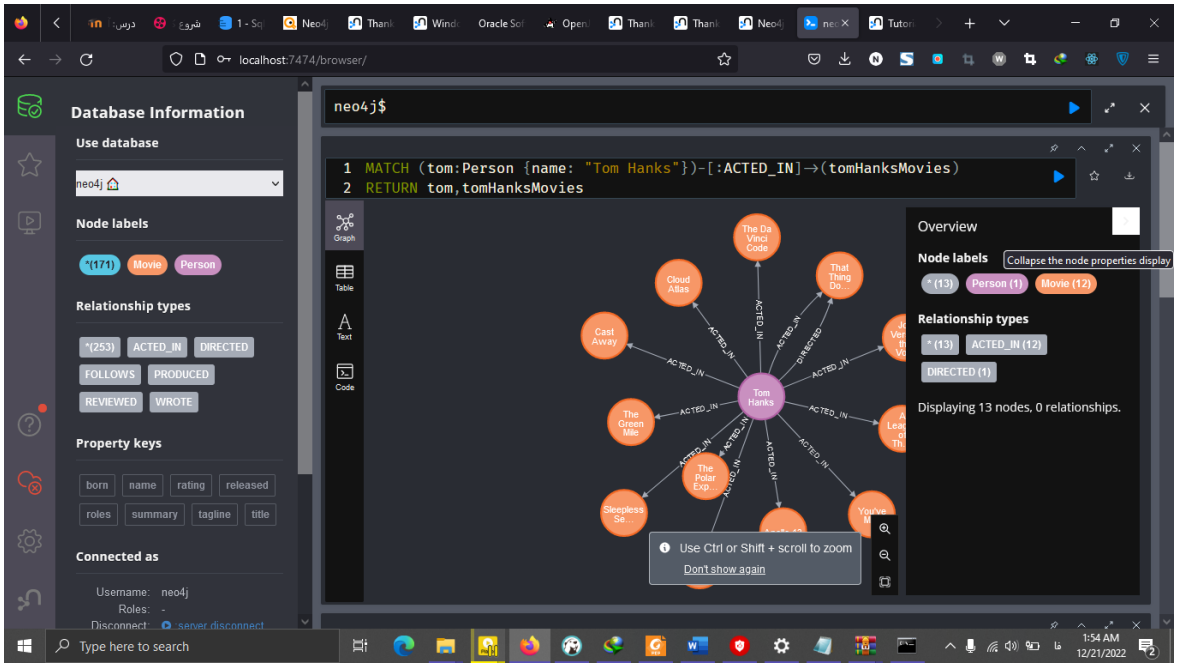


پیدا کردن فیلم هایی که در release 1990 شده اند.



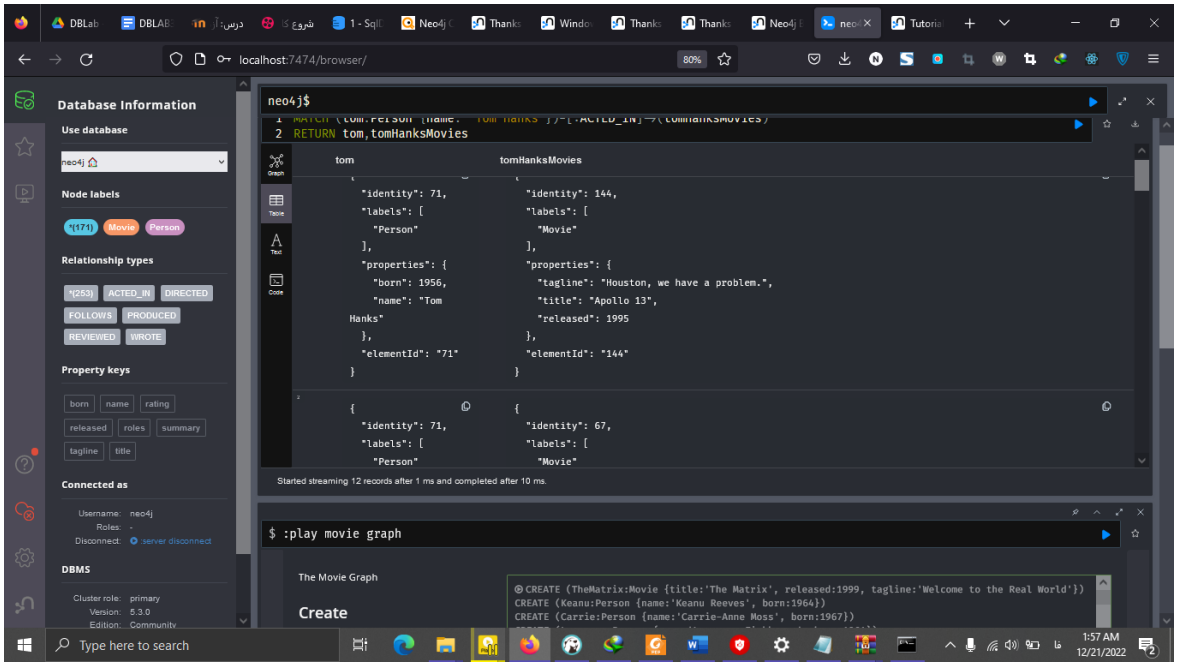
این query تمام نودهای movie که سال تولیدشان (relased)بین 1990 و 2000 است را پیدا کرده و تایتل آن ها را return میکند.

لیست تمام فیلم های تام هنکس

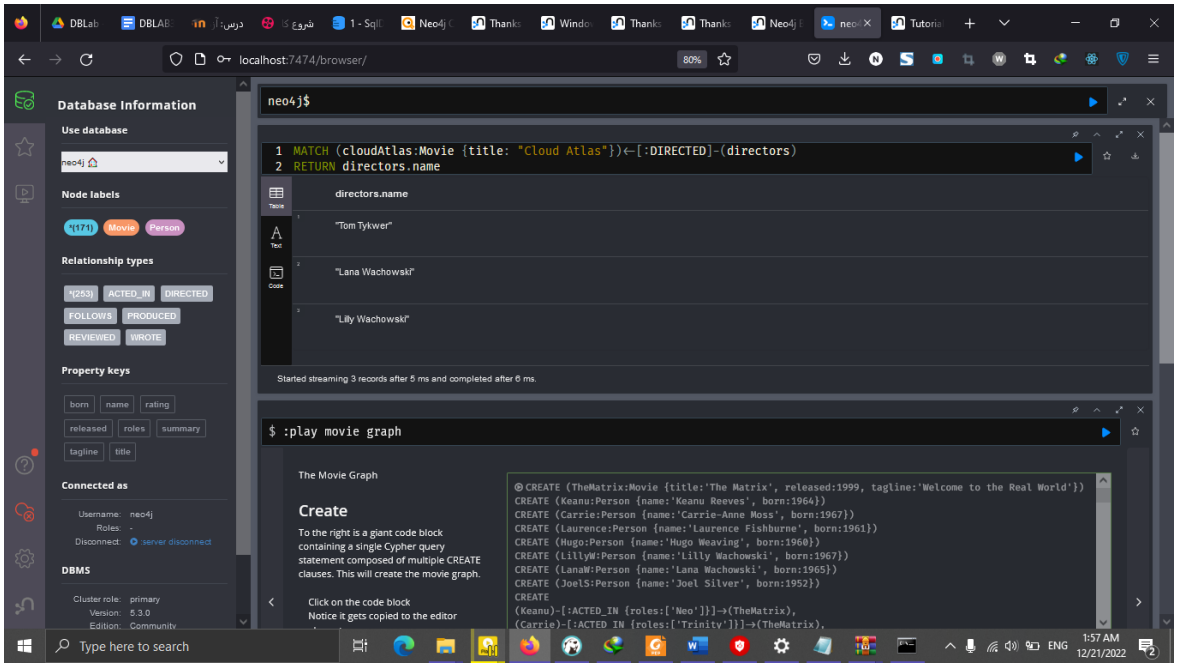


در این query نودهای person که نام tom hanks دارند و همچنین تمام نودهایی که با آن رابطه acted_in را دارند باز میگردانیم.(فیلم هایی که در آن ها بازی کرده) نود تام هنکس و نودهای فیلم باهم رابطه مستقیم دارند.

جدول زیر هم نتیجه را نشان میدهد

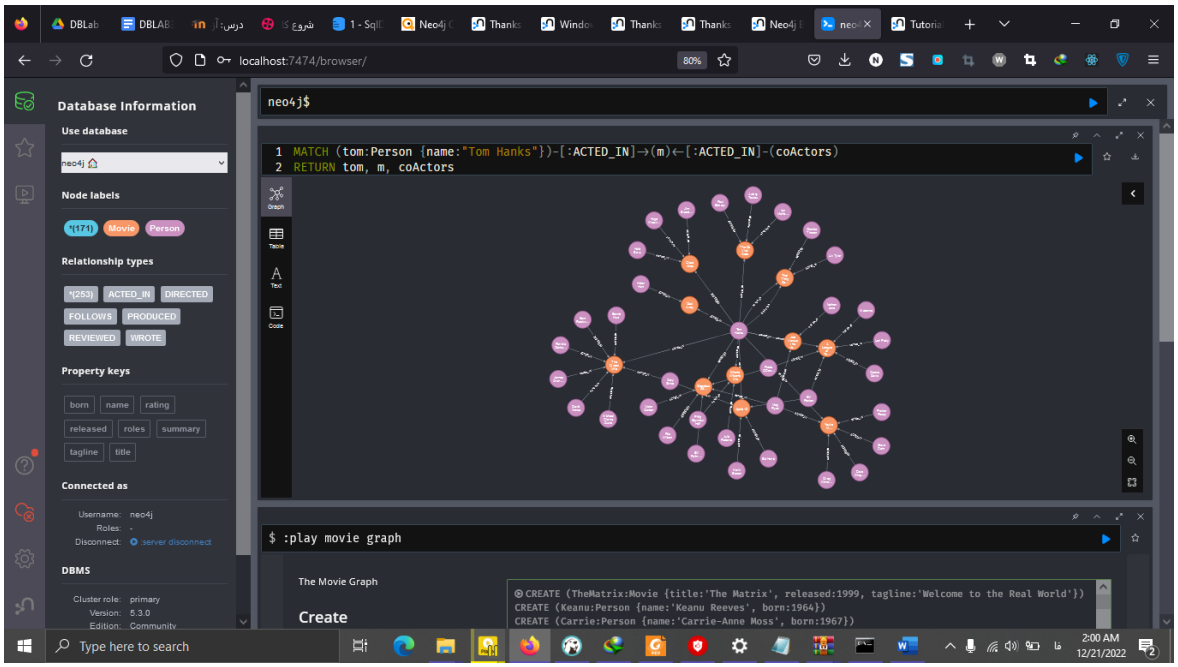


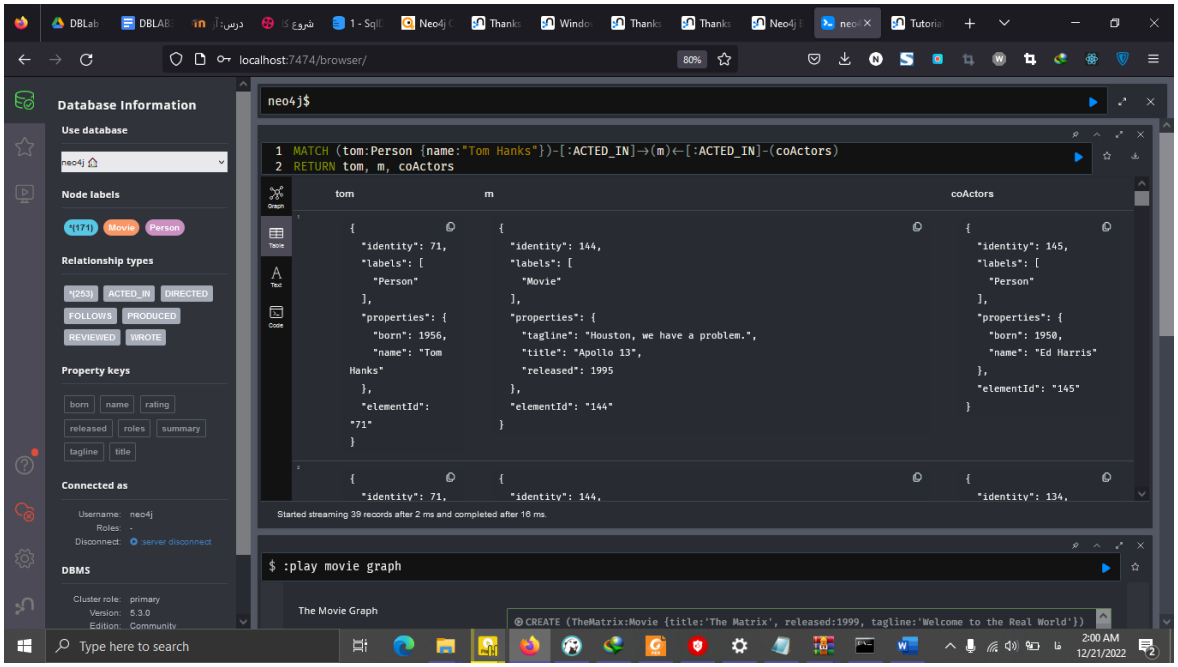
لیست کارگردان های فیلم Cloud Atlas



این دستور اسم تمام نودهایی که رابطه directed با نود movie که نامش cloud atlas است دارد را بازمیگرداند.

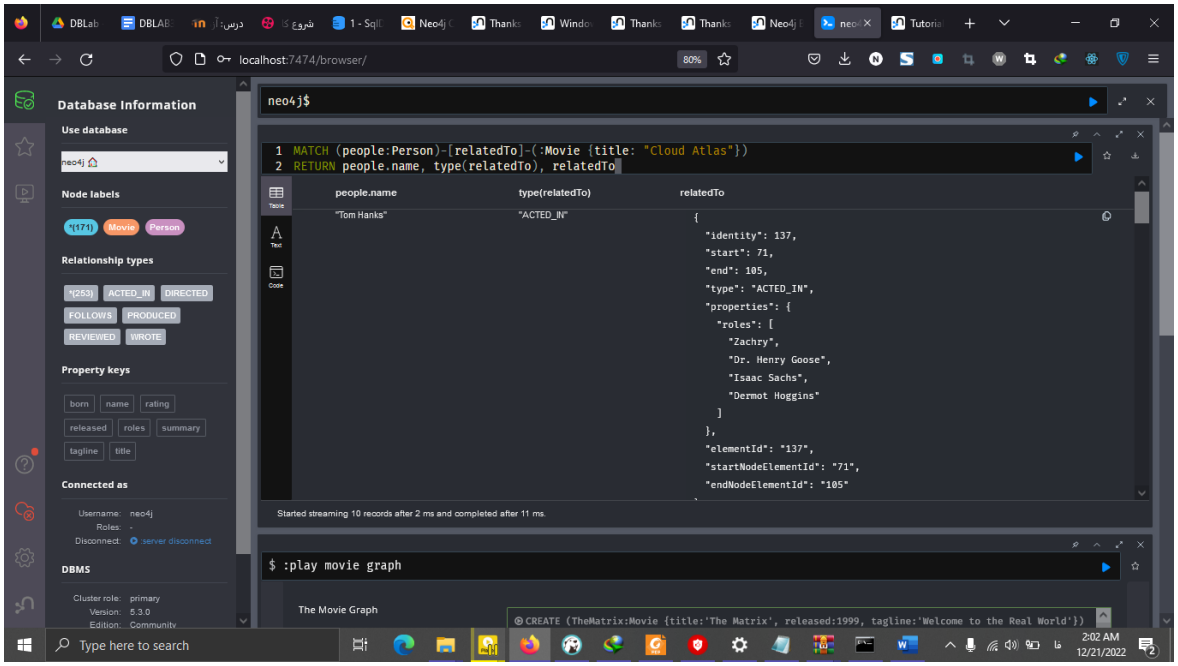
تمام فیلم هایی که تام هنکس در آن ها بازی کرده و تمام بازیگرانی که در این فیلم ها بازی کرده اند.





در این query نودهایی که با تام هنکس رابطه acted_in دارند و همچنین هر نود دیگه ای که با این فیلم ها رابطه acted_in دارد را میگیریم.

نام تمام افرادی که به فیلم cloud atlas ارتباط دارند و نوع ارتباطشان.

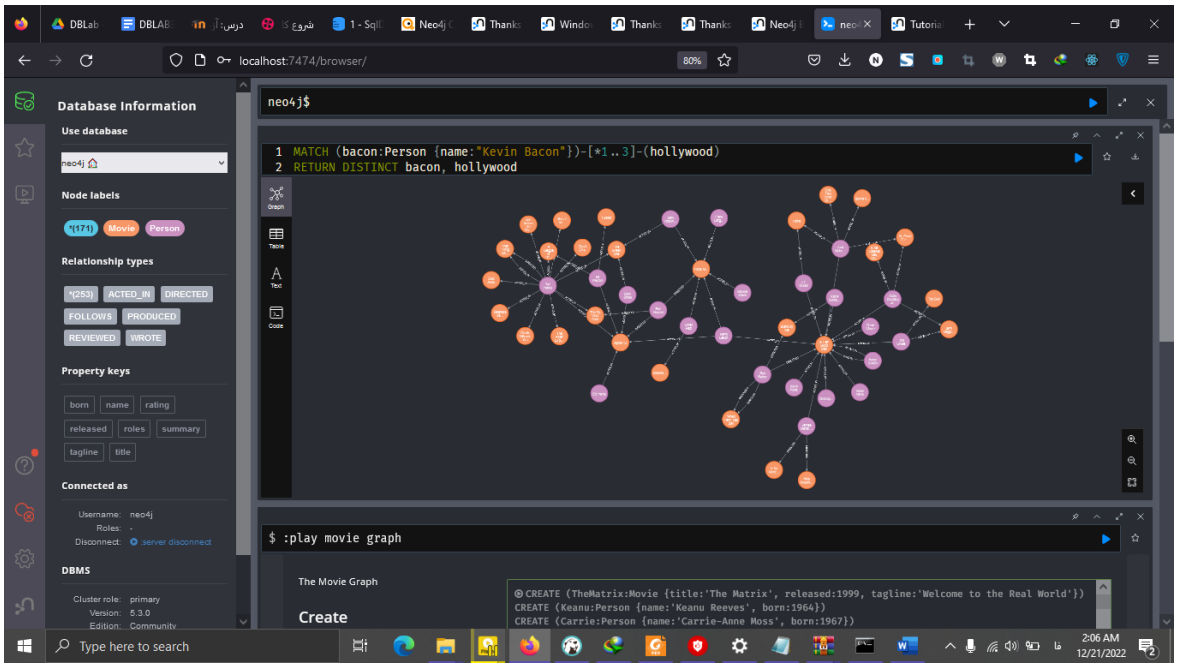


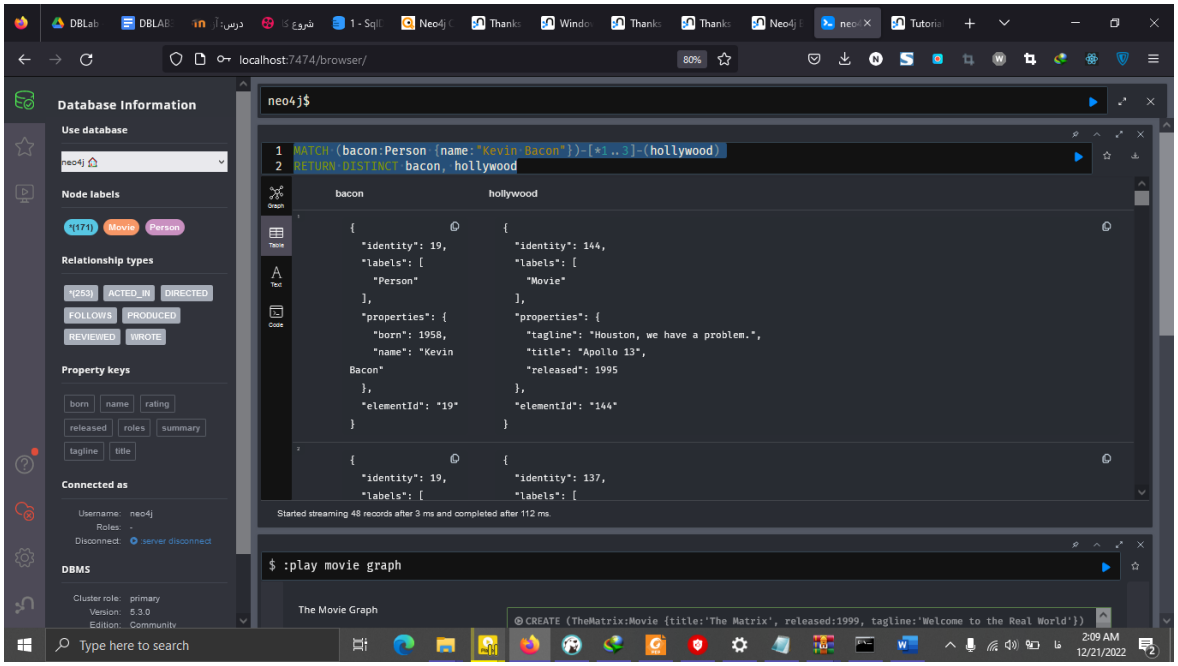
تمام نام نودهایی که رابطه relatedTo با فیلمی که تایتلcloud atlas دارد دارند و نوع رابطه شان را با استفاده از typeبازمیگردانیم.

Query که در آن هر نودی که در گراف فاصله ای بین 1 تا 3 دارد از نود Kevin Bacon را برمیگردانیم. (چه فیلم باشد چه person) فاصله 1 تا 3 را با

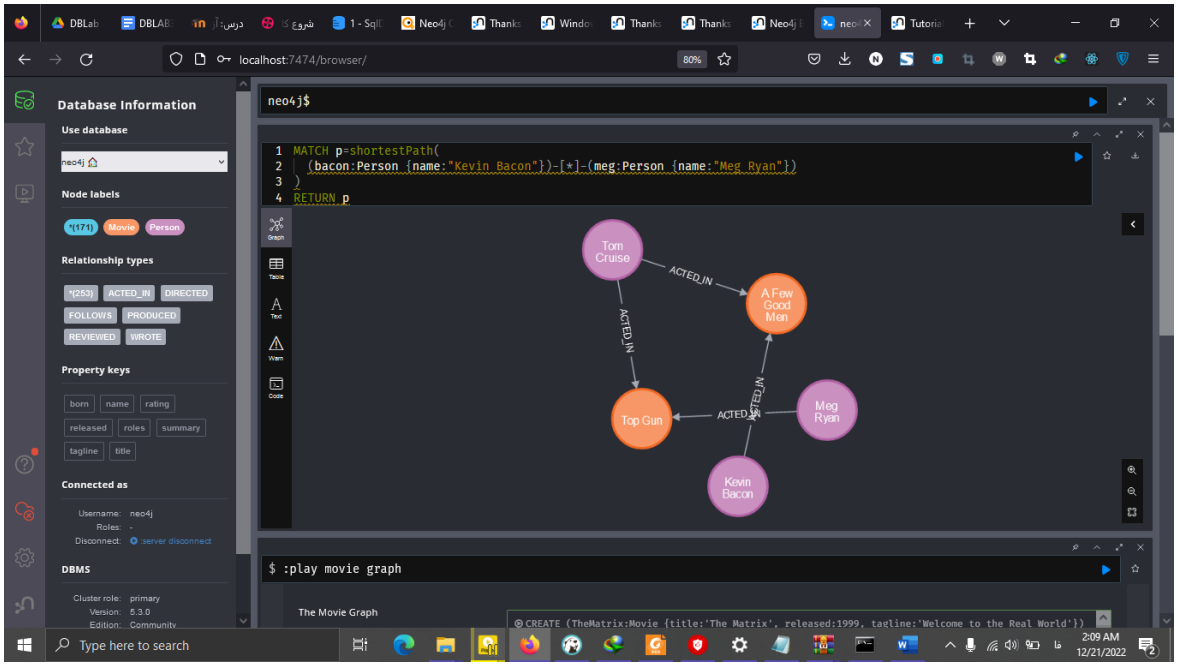
```
[*1..3]
```

نشان میدهیم.



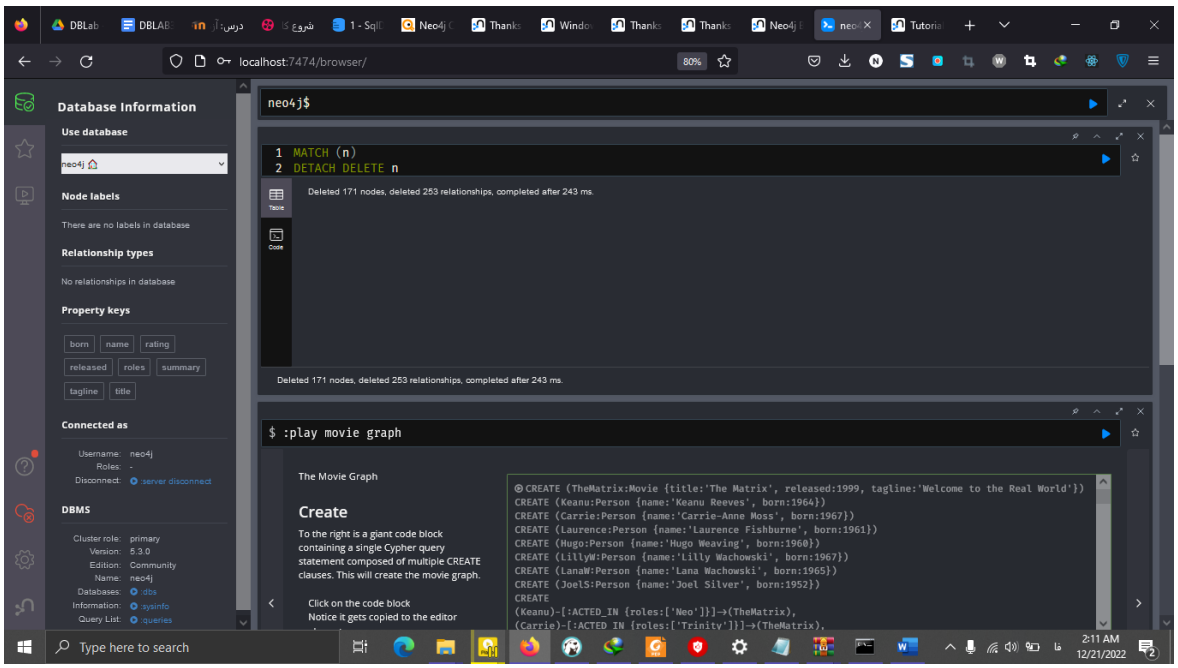


کوتاه ترین مسیر بین دو نود Kevin Bacon , Meg Ryan person را بازمیگردانیم که ممکن است یک نود و یا یک رابطه باشد.

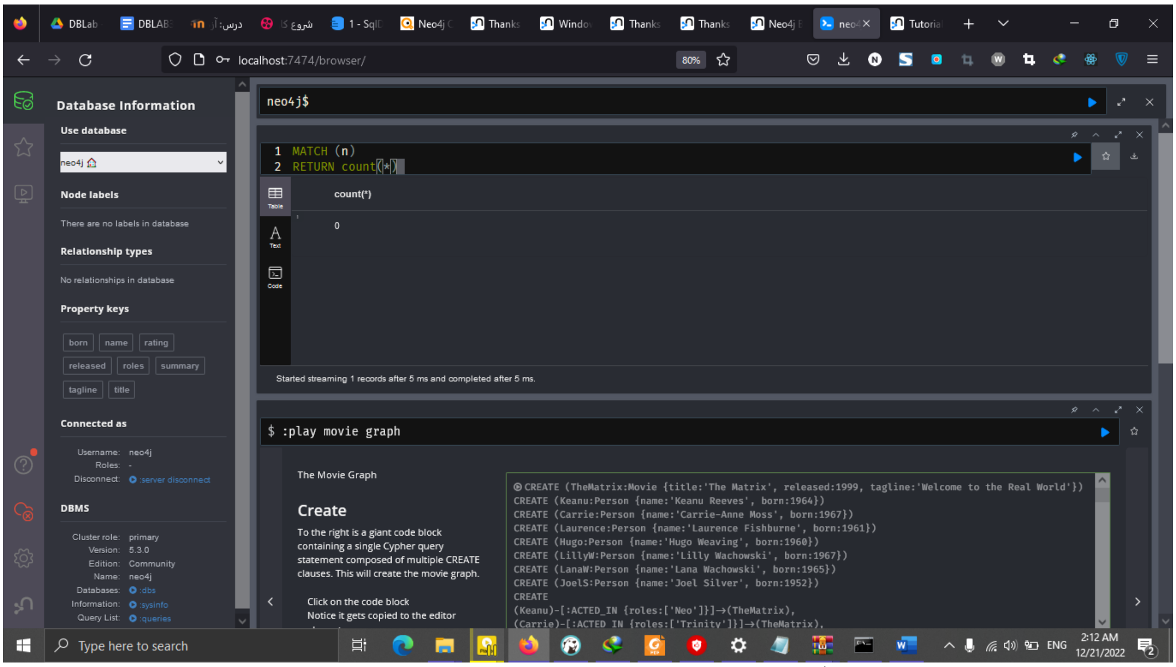


ابتدا دو نود person مورد نظر را پیدا میکنیم و سپس مسیر های بین این دو نود را پیدا میکنیم (تمام مسیر ها *) و در انتها shortest path این مسیر ها را در نظر میگیریم.

بعد از اتمام کارمان این دیتاست را پاک میکنیم با استفاده از دستور زیر

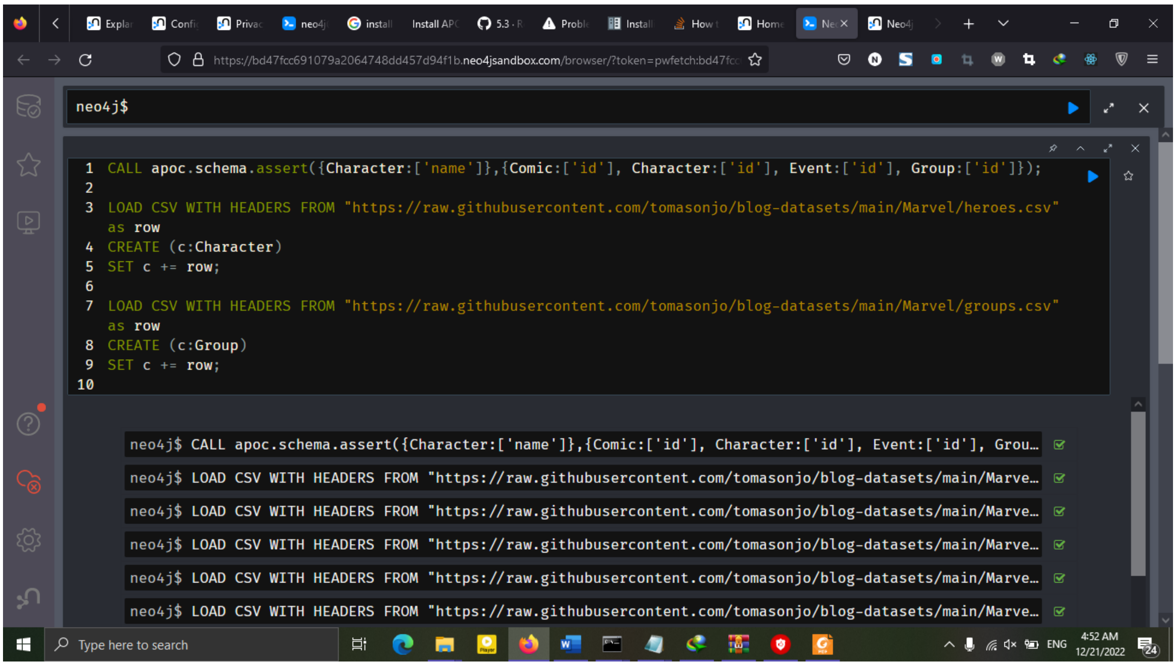


یک دور چک میکنیم که دیتاست پاک شده باشد

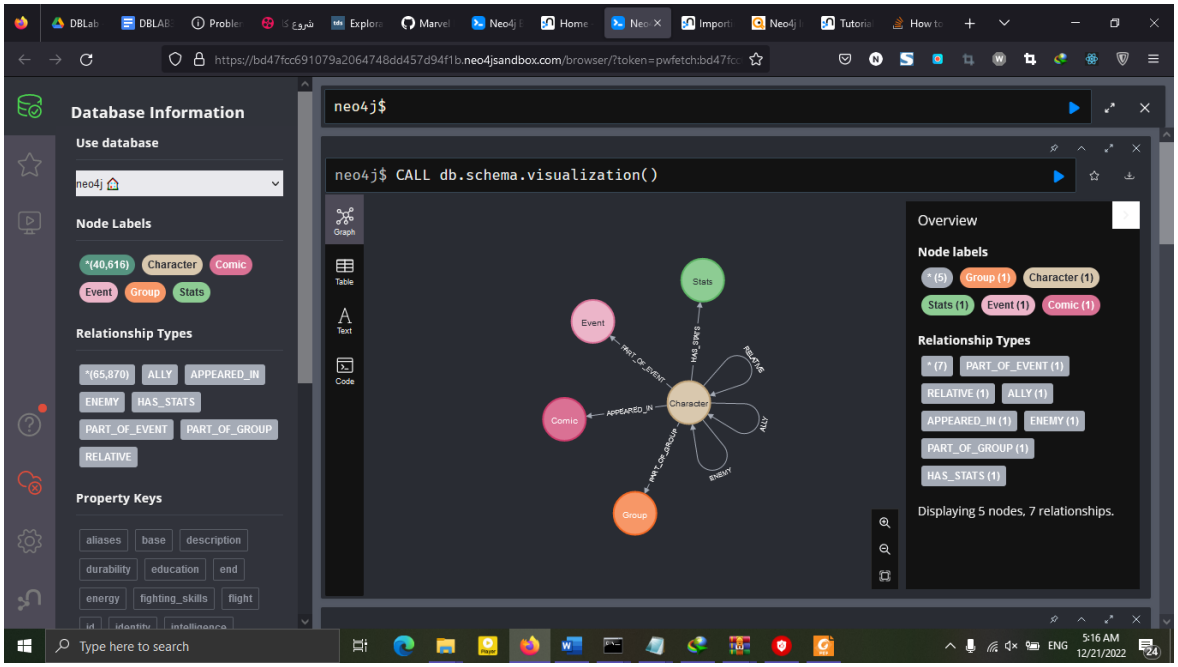


همانطور که مشاهده میکنید 0 باز میگرداند.

لود کردن دیتاست مارول. طبق دستورات موجود در مقاله دیتا را لود میکنیم.



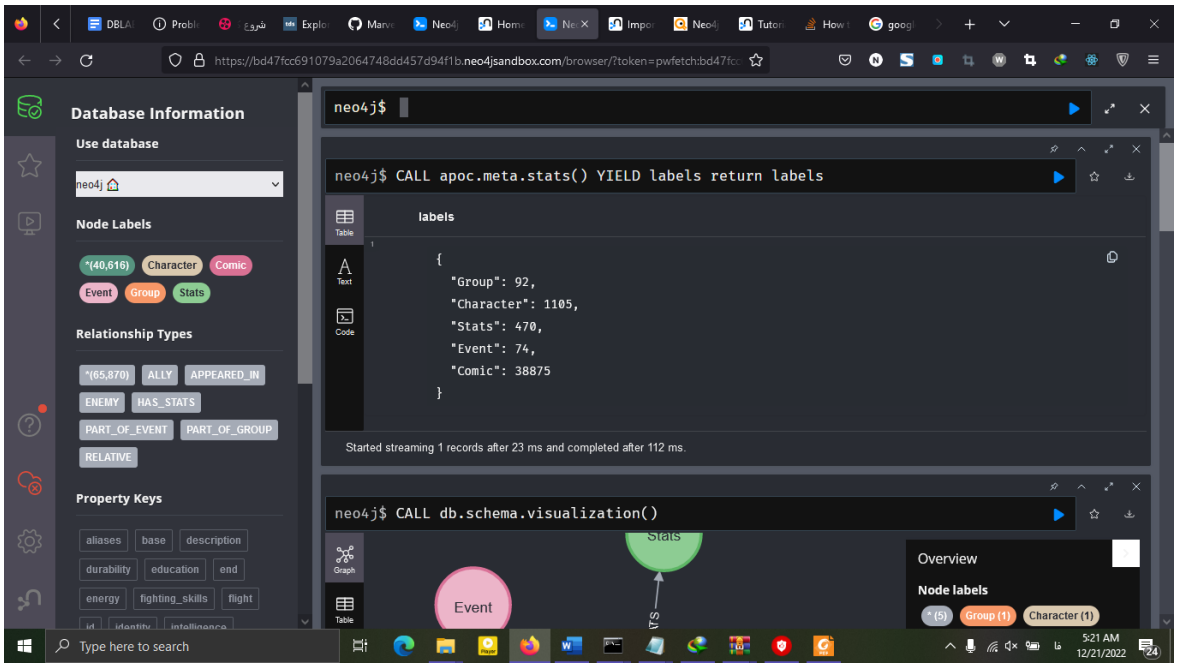
نمایش شمای گراف



با دستور call یک procedure را صدا میکنیم

در مرکز characters وجود دارد که می تواند جزئی از یک event باشد یا در چندین کامیک ظاهر شود ، یا به یک گروه تعلق داشته باشد و یا آمار داشته باشد. کاراکترها می توانند دشمن هم، متحد و یا فامیل باشند.

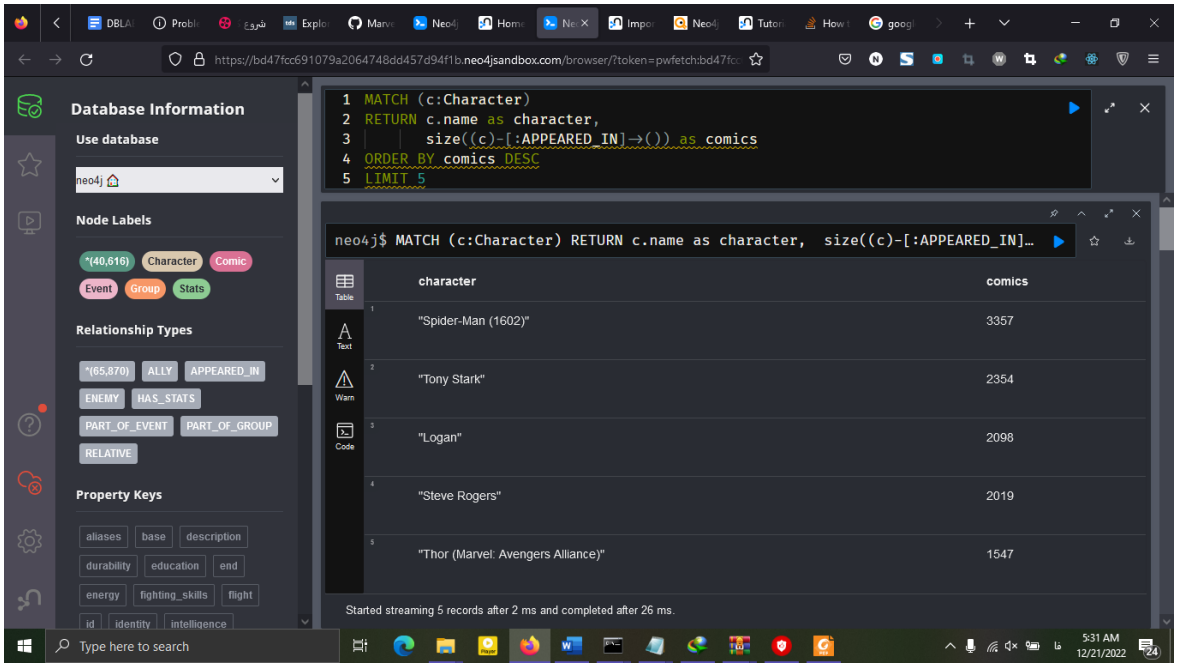
برای درک بهتر سایز گراف این دستور را اجرا میکنیم.



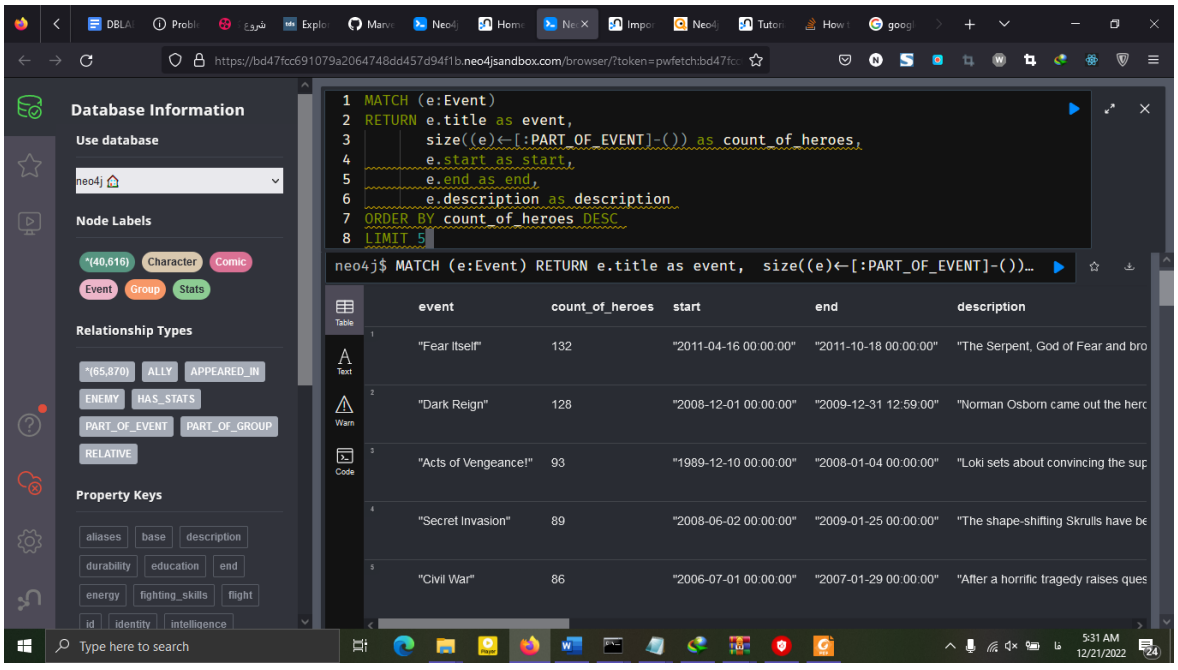
درکل 1105 کاراکتر وجود دارد که در 338875 کامیک ظاهر شده است. 470 کاراکتر آمار دارند. 92 گروه 74 ایونت در گراف ذخیره شده اند.

برای تحلیل گراف تحلیل خود را با تعداد کاراکترهایی که بیشترین تکرار را در کامیک دارند شروع میکنیم.

کوئری زیر تعداد کاراکتری که برای کامیک ظاهر شده اند را بر اساس تعداد رابطه appeared_in را با دستور size پیدا میکند (تعداد روابط که نود کاراکتر آن ها کاراکتر مورد نظر است)و آن ها را مرتب کرده و 5 تای اول را باز میگرداند.

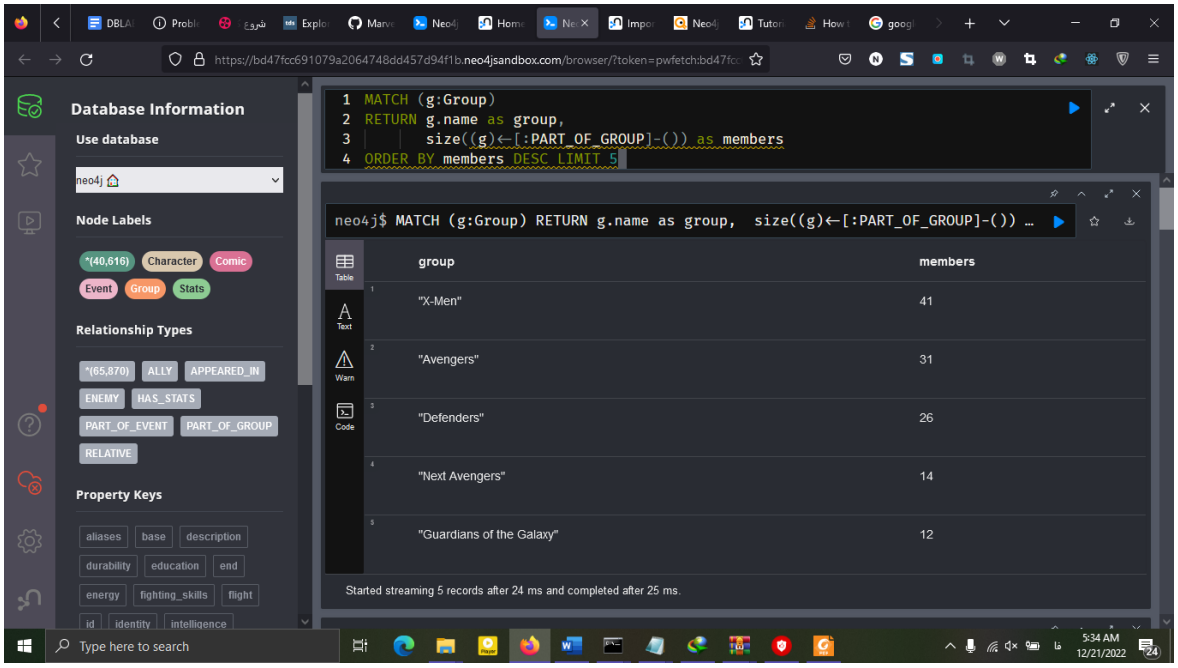


با استفاده از کوئری زیر رویدادهایی را پیدا میکنیم که بیشترین کاراکتر را دارند.



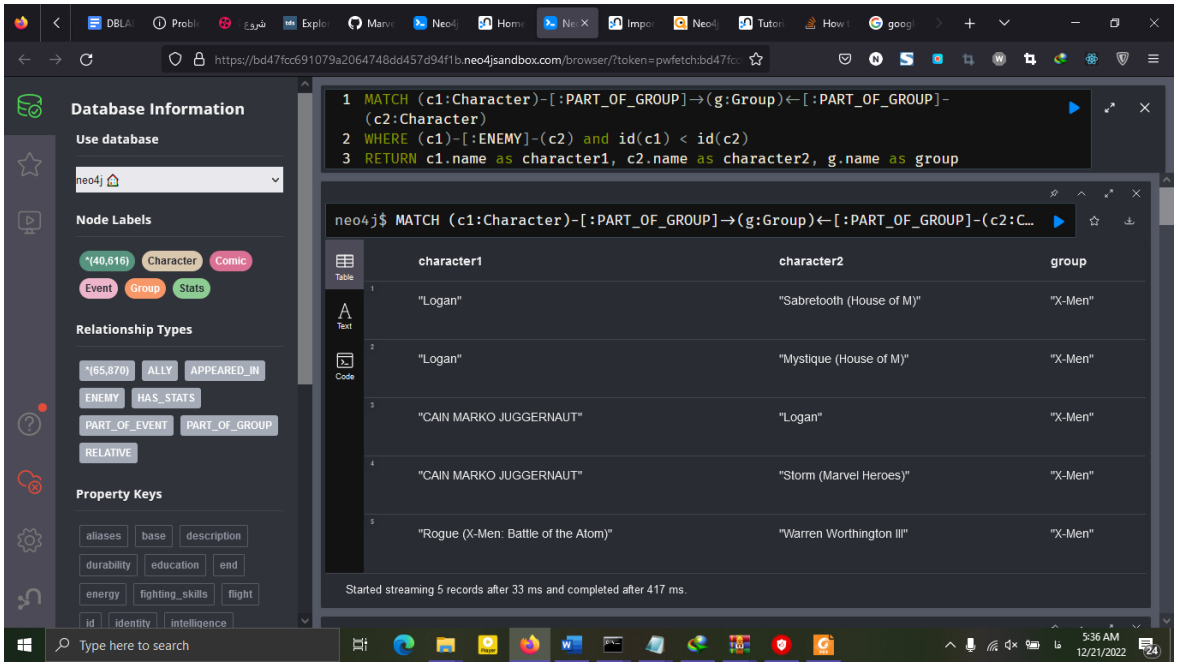
در این کوئری برای هر رویداد تعداد نودهایی که به آن رابطه part_of_event دارند را با دستور size میگیرد و براساس تعداد نودها مرتب کرده و 5 تای اولی را برمیگرداند.

پرجمعیت ترین گروه ها



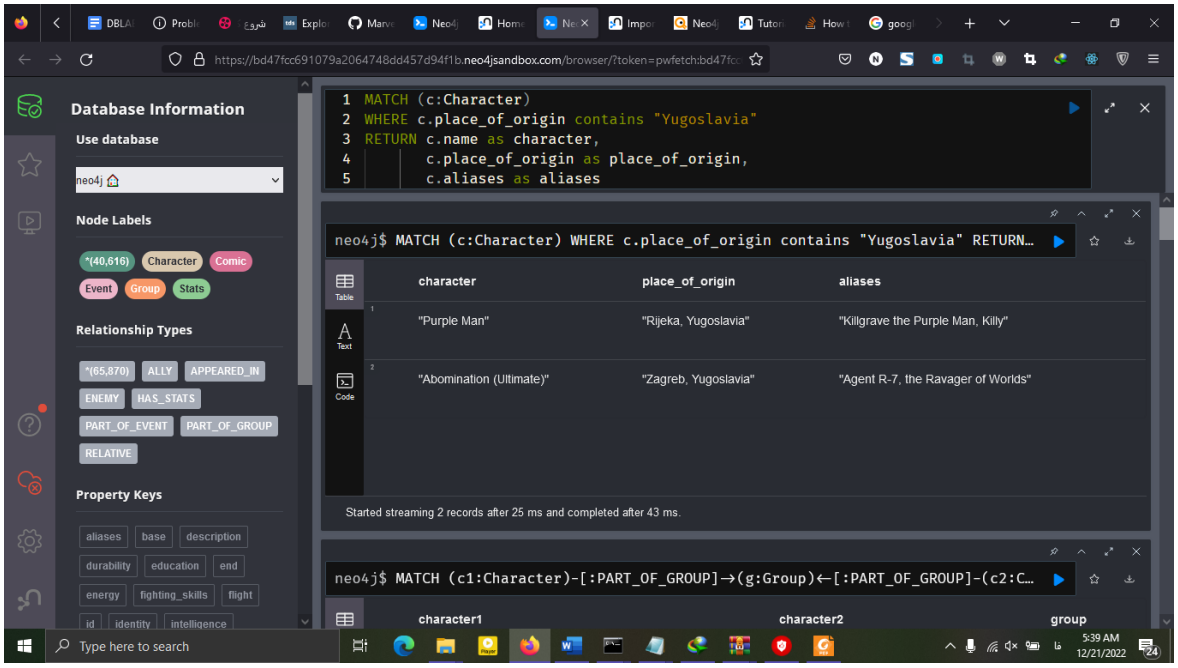
در این کوئری برای هر گروه تعداد نودهایی که به آن رابطه part_of_group دارند را با دستور size میگیرد و براساس تعداد نودها مرتب کرده و 5 تای اولی را برمیگرداند.

کاراکترهایی که باهم دشمن اند ولی عضو یک گروهند.



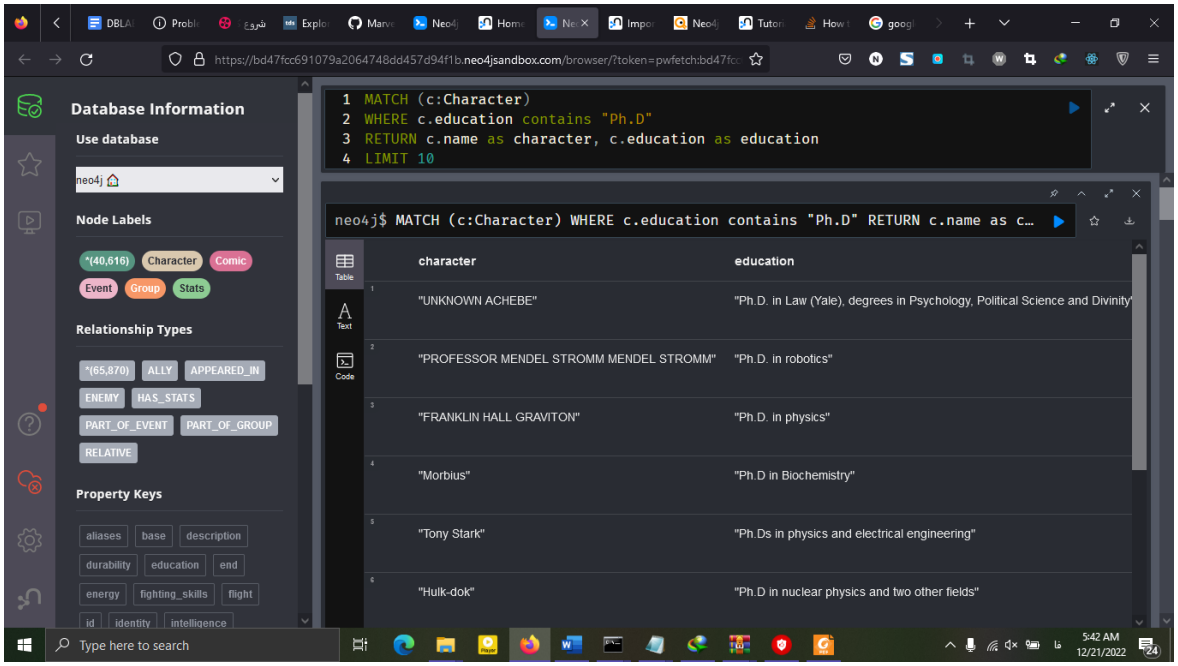
ابتدا هر دو کاراکتر c1, c2 ای را میابیم که عضو یک گروه باشند و سپس چک میکنیم که این دو یال دشمن داشته باشند و همینطور چک میکنیم که یکبار بیایند در جدول.

کاراکترهایی که از Yugoslavia هستند.



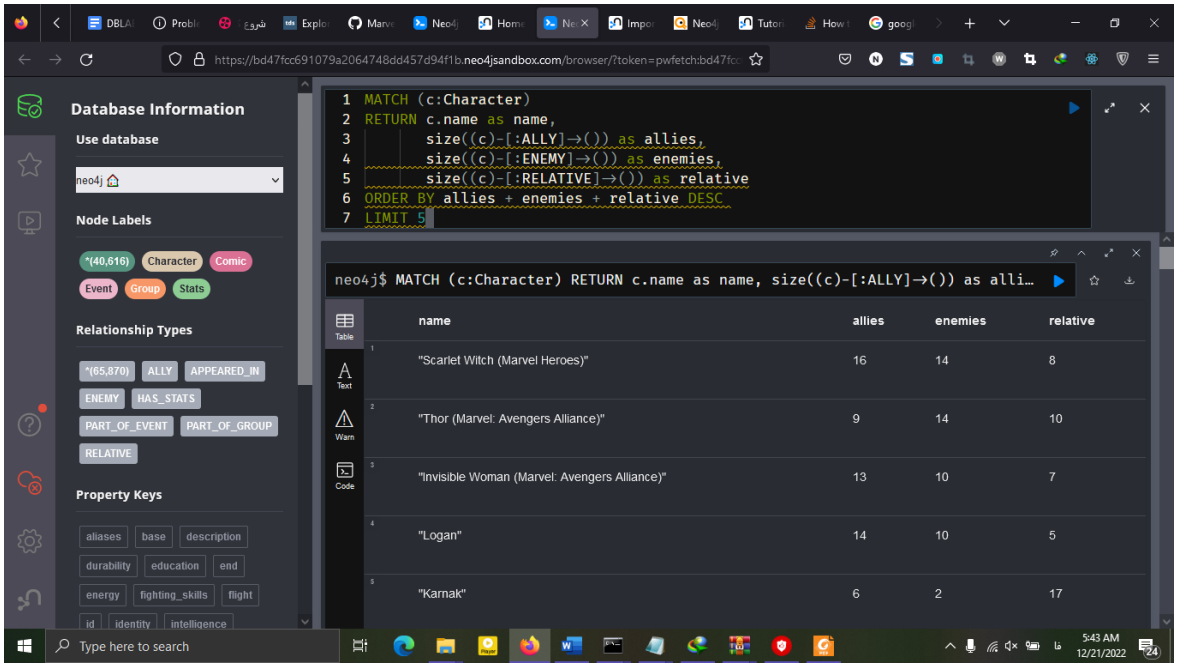
ابتدا چک میکنیم که محل تولد شامل Yugoslavia باشد سپس نامو محل تولد و لقب های آن کاراکتر را برمیگردانیم.

کاراکترهایی که PHD دارند



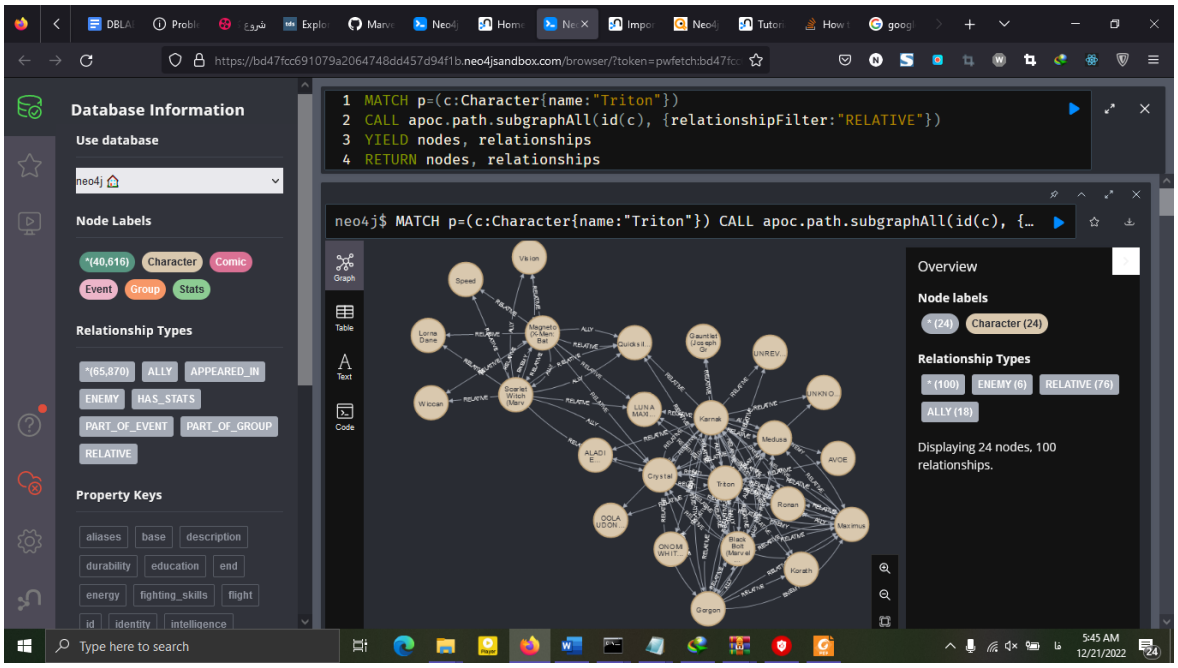
چک میکنیم که کاراکتر شامل Ph.D باشد و نام و تحصیلات را برمیگردانیم با محدودیت 10 تا

کاراکترهایی که بیشترین رابطه را با بقیه دارند.



تعداد نودها را براساس تعداد یال های Ally, enemy, relative که با size به دست اوردیم میگیریم و براساس مجموعشان مرتب میکنیم و 5 تای اول را برمیداریم.

گراف ارتباط خانواده Triton با یکدیگر



با استفاده از procedure subgraphAll گرافی با تمام روابط با کاراکتر Triton را میسازیم. این تابع 2 ارگومان میگیرد اولی ایدی یک نود و دومی رابطه خواسته شده است. این تابع تمام نودهایی که با ان نود اولیه رابطه خواسته شده را دارند برمیگرداند.

مشکلات و توضیحات تکمیلی

برای حل کانفیگ کتابخانه apoc خیلی به مشکل برخوردم.

آنچه آموختم / پیشنهادات

در این دستورکار به noe4j آشنا شدم که بسیار جالب بود و استفاده از داکیومنت خیلی خوب بود.