دانشگاه فردوسی مشهد
دانشکده‌ی مهندسی

پروژه‌ی کارشناسی
مهندسی کامپیوتر

عنوان:

ایجاد گراف دانش با Neo4j

نگارش:

سارا اسدی

استاد راهنما:

دکتر محسن کاهانی

استاد مشاور:

مریم کرابی

زمستان ۱۴۰۱

سپاس

از استاد بزرگوارم، دکتر کاهانی، که با کمک ها و راهنمایی های بی دریغشان، بنده را در انجام این پروژه یاری داده‌اند، تشکر و قدردانی میکنم.

از خانم مریم کرابی که در طول مدت انجام این پروژه، زمان و علم و دانش زیادی را در اختیار من قرار داد نیز بسیار سپاسگزارم.

از جناب دکتر اله بخش که زحمت داوری این پروژه را به عهده داشته اند نیز کمال تشکر را دارم.

همچنین تشکر میکنم از دوستان و همکلاسیان گرامی که در طی دوران تحصیل در کنار من بوده اند.

و در آخر از پدر، مادر و خواهرانم که در همواره در طی این سالیان پشتیبان من بوده اند.

چکیده

با رشد تصاعدی داده ها، نیاز به مدیریت کارآمد داده ها و سیستم های بازیابی اهمیت فزاینده ای پیدا کرده است. گراف‌های دانش به عنوان یک راه حل امیدوارکننده برای مقابله با این چالش با نمایش داده ها به شیوه ای ساختاریافته و به هم پیوسته پدیدار شده اند. در این پایان نامه، فرآیند ایجاد یک گراف دانش از منابع داده بدون ساختار را بررسی می کنیم و بر جنبه های عملی توسعه چنین سیستمی تمرکز می کنیم. ما با بحث در مورد معرفی گراف دانش و هستی‌شناسی و اهمیت آنها در زمینه مدیریت داده ها شروع می کنیم. هدف اصلی این پایان نامه توسعه یک گراف دانش برای یک حوزه خاص و نمایش اثربخشی آن در سازماندهی و استخراج اطلاعات سودمند از مجموعه داده‌های بزرگ است. ما از یک مجموعه داده دنیای واقعی برای نشان دادن اثربخشی گراف دانش استفاده خواهیم کرد که برای بنیاد امام رضا (ع) می‌باشد. در ایجاد این گراف دانش مراحلی از جمله جمع آوری، پیش پردازش داده و شناسایی موجودیت، استخراج رابطه و تولید گراف نهایی ذکر شده است. امیدواریم که این پژوهش به مجموعه دانش رو به رشد در این زمینه کمک کند و الهام بخش کار آینده برای محققان و متخصصان علاقه مند به استفاده از نمودارهای دانش برای مدیریت و تجزیه و تحلیل داده ها باشد.

کلیدواژه ها: Neo4j، گراف دانش، پایگاه داده گرافی، هستی شناسی .

فهرست مطالب

فهرست شکل‌ها

فصل ۱

مقدمه

در سال‌های اخیر، حجم داده‌های تولید شده توسط مشاغل، افراد و ماشین‌ها افزایش بسیار یافته و به پدیده‌ای منجر شده که معمولاً به عنوان پدیده کلان داده شناخته می‌شود. این داده‌ها می‌توانند از طیف گسترده‌ای از منابع، از جمله پلتفرم‌های رسانه‌های اجتماعی، دستگاه‌های IoT ، تراکنش‌های مالی و غیره به دست آیند، و اغلب بدون ساختار یا نیمه‌ساختارمند هستند، که مدیریت و تجزیه و تحلیل با استفاده از ابزارهای مدیریت داده سنتی را دشوار می‌کند. حجم زیاد و پیچیدگی کلان داده چالش‌هایی را ایجاد می‌کند، از جمله:

- ذخیره سازی و بازیابی داده ها: سیستم های ذخیره سازی سنتی داده ها ممکن است توانایی مدیریت حجم داده های تولید شده توسط منابع بزرگ داده را نداشته باشند، و ذخیره و بازیابی به موقع داده ها را دشوار می‌کند.
- پردازش و تجزیه و تحلیل داده‌ها: با وجود داده‌های زیادی برای پردازش، ابزارهای سنتی تجزیه و تحلیل داده‌ها ممکن است برای ارائه بینش و هوش عملی در مدت زمان معقول با مشکل مواجه شوند.
- کیفیت داده: داده‌های بزرگ می‌توانند کثیف باشند، با داده‌های ناسازگار یا ناقص از منابع متعدد. این می تواند دریافت یک تصویر کامل و دقیق از آنچه داده ها به ما می گویند دشوار باشد.
- حریم خصوصی و امنیت داده ها: با جمع آوری و ذخیره داده های بیشتری، نگرانی هایی در مورد حفظ حریم خصوصی و امنیت داده ها وجود دارد، از جمله خطر دسترسی غیرمجاز یا نقض

داده‌ها.

عدم استفاده از تجزیه و تحلیل داده ها برای مشاغل و سازمان ها می تواند منجر به مشکلات متعددی مانند عدم شناسایی ناکارآمدی ها، ناتوانی در رقابت، از دست دادن مشتریان، تصمیمات ضعیف، از دست دادن فرصت های بهبود و ... شود که موجب هدر رفتن منابع، زمان، پول و عواقب منفی بسیاری گردد.

برای مقابله با این چالش ها، کسب و کارها و سازمان ها به فناوری های جدیدی مانند پایگاه های داده NoSQL روی آورده اند که برای مدیریت حجم زیادی از داده های بدون ساختار یا نیمه ساختار یافته طراحی شده اند. Neo4jیکی از این فناوری‌هاست، و چندین مزیت کلیدی را نسبت به پایگاه‌های داده سنتی و دیگر پایگاه‌های داده NoSQL در مدیریت و تجزیه و تحلیل داده‌های بزرگ ارائه می‌دهد. اول و مهمتر از همه، مدل پایگاه داده گراف Neo4jبه ویژه برای برنامه های کاربردی کلان داده مناسب است، زیرا امکان مدیریت کارآمد و پرس و جو از مجموعه داده های بسیار به هم پیوسته را فراهم می‌کند. این امکان استخراج بینش از روابط پیچیده بین نقاط داده را فراهم می کند که ممکن است به راحتی با استفاده از ابزارهای مدیریت داده سنتی قابل شناسایی نباشند. علاوه بر این، زبان پرس و جوی آن یعنی Cypher، به گونه ای طراحی شده است که بصری و آسان برای استفاده باشد و به توسعه دهندگان و تحلیلگران داده اجازه می دهد تا به سرعت و به راحتی پرس و جوهای پیچیده را در مجموعه داده های بزرگ انجام دهند. و از آنجایی که Neo4jبسیار مقیاس پذیر است، به راحتی می تواند رشد کند و با نیازهای کسب و کارها و سازمان ها سازگار شود زیرا نیازهای داده آنها در طول زمان تکامل می یابد. به طور کلی، نقاط قوت Neo4jدر مدل‌سازی داده‌ها، پرس و جو و مقیاس‌پذیری، آن را به ابزاری قدرتمند برای مدیریت و تجزیه و تحلیل داده‌های بزرگ تبدیل می‌کند و آن را به گزینه‌ای ایده‌آل برای کسب‌وکارها و سازمان‌هایی تبدیل می‌کند که به دنبال کسب بینش از مجموعه داده‌های پیچیده و بسیار به هم مرتبط هستند.

در ادامه به بیان شرح کل صورت پروژه، دلایل انجام و بررسی ساختار کل این گزارش میپزداریم.

۱-۱ تعریف پروژه

پروژه شامل چهار بخش کلی میباشد که عبارت اند از:

۱. استخراج موجودیت ها و روابط از داده های جمع آوری شده از فایل OWL

۲. ساخت گراف دانش از تمام کلاس های موجود و تهیه یک طرح جامع

۳. ایجاد پایگاه داده گرافی بر اساس داده های پروژه بنیاد امام رضا و مقادیر گردآوری شده از سایت

۴. طراحی یک رابط کاربری مناسب برای نمایش خروجی پروژه

۱-۲ اهداف پروژه

هدف از انجام این پروژه استفاده از پایگاه داده گرافی و ایجاد گراف دانش برای داده های غیر ساختار یافته حوزه خاص و نمایش اثربخشی آن در سازماندهی و استخراج اطلاعات سودمند از مجموعه داده های بزرگ است. ما از یک مجموعه داده دنیای واقعی برای نشان دادن اثربخشی نمودار دانش استفاده خواهیم کرد که برای بنیاد امام رضا (ع) می‌باشد. در ایجاد این گراف دانش مراحلی از جمله جمع آوری، پیش پردازش داده و شناسایی و استخراج موجودیت و روابط و تولید گراف نهایی شرح داده شده است.

۱-۳ ساختار پروژه

این گزارش شامل پنج فصل می‌باشد که شامل محتوی زیر می باشد. فصل اول (این فصل) شامل تعریف کلی صورت پروژه و بیان کلیات می باشد. فصل دوم دربرگیرنده معرفی گراف دانش_ آشنایی با OWL و تفاوت آن با گراف دانش می باشد. در فصل سوم روش پیشنهادی برای انجام پروژه و پیاده سازی به صورت مفصل شرح داده می شود. سپس در فصل چهارم به نتایج به دست آمده از این عملیات اشاره میکنیم. فصل پنجم شامل خلاصه و جمع بندی پروژه به همراه کارهای پیش رو می باشد که به بیان چشم انداز ما برای ادامه پروژه و اقدامات قابل انجام توسط دیگران می پردازد.

فصل ۲

پیش زمینه پروژه و تعاریف کلی

در این بخش به توضیح برخی موارد و اصطلاحاتی که در طول گزارش استفاده شده است و ممکن است برای همگان ناآشنا باشد می‌پردازیم.

۲-۱ گراف دانش چیست؟

گراف دانش نوعی پایگاه داده است که دانش را به عنوان مجموعه ای از موجودیت های به هم پیوسته و روابط آنها با یکدیگر نشان می‌دهد. گراف‌های دانش معمولاً حاوی اطلاعاتی در مورد حوزه‌ها یا حوزه‌های موضوعی خاص، مانند مفاهیم علمی، فرآیندهای تجاری یا شبکه‌های اجتماعی هستند و یک ابزار قدرتمند برای نمایش و تجزیه و تحلیل داده های پیچیده ارائه می‌کنند و به طور فزاینده‌ای در طیف گسترده ای از کاربردها در صنایع مورد استفاده قرار می گیرند. در گراف دانش، موجودیت ها به صورت گره و روابط بین موجودیت ها به صورت لبه ها یا پیوندها نمایش داده می شوند. به عنوان مثال، در یک نمودار دانش از افراد و روابط آنها، هر فرد به عنوان یک گره نشان داده می شود، و روابط آنها (مانند عضو خانواده، دوستان و همکاران) به عنوان یال‌هایی نشان داده می‌شود که گره ها را به هم متصل می کنند. استفاده از نمودار دانش امکان نمایش انعطاف‌پذیرتر و ظریف‌تر داده‌ها را نسبت به پایگاه‌های داده سنتی فراهم می‌کند. از نمودارهای دانش می توان برای ثبت پیچیدگی و به هم پیوستگی سیستم های دنیای واقعی استفاده کرد و امکان پرس و جو و تحلیل کارآمدتر حجم زیادی از داده ها را فراهم کرد و به ویژه برای برنامه هایی که نیاز به استدلال و استنباط دارند، مانند پردازش زبان طبیعی، سیستم های توصیه

و کشف تقلب مناسب هستند. گراف‌های دانش معمولاً با استفاده از فناوری های وب معنایی، مانند مدل داده RDF (چارچوب توصیف منابع) و OWL (زبان هستی شناسی وب) برای نمایش هستی شناسی ها و ساختارهای دانش ساخته می شوند. علاوه بر گره‌ها و روابط، نمودارهای دانش ممکن است دارای ویژگی‌ها یا ویژگی‌هایی باشند که موجودیت‌ها را با جزئیات بیشتری توصیف می‌کنند. به عنوان مثال، یک گره شخصی در یک نمودار دانش ممکن است ویژگی هایی مانند سن، شغل و علایق داشته باشد. گراف دانش را می توان برای نمایش طیف وسیعی از انواع داده ها، از جمله داده های ساخت یافته (مانند داده های پایگاه داده های رابطه ای) و داده های بدون ساختار (مانند متن و محتوای چند رسانه ای) استفاده کرد و می‌توان با استفاده از تکنیک‌های مختلف، از جمله الگوریتم‌های نمودار، پردازش زبان طبیعی، و یادگیری ماشین، پرس و جو و تحلیل کرد. یکی از مزیت‌های کلیدی نمودارهای دانش، توانایی آن‌ها در یکپارچه‌سازی داده‌ها از منابع متعدد، مانند مجموعه‌های داده عمومی، پایگاه‌های داده داخلی و پلتفرم‌های رسانه‌های اجتماعی است. برخی از شناخته شده ترین نمودارهای دانش عبارتند از نمودار دانش گوگل که برای ارائه پرس و جو و پاسخ در نتایج جستجو استفاده می شود و نمودار دانش DBpedia که از ویکی پدیا مشتق شده و حاوی اطلاعات ساختاریافته درباره افراد، مکان ها و چیزها است. استفاده از نمودارهای دانش به طور فزاینده ای در زمینه هایی مانند علم داده، هوش مصنوعی و مدیریت اطلاعات رایج می شود، زیرا سازمان ها به دنبال به دست آوردن بینش عمیق تر در مورد داده های خود و استفاده بهتر از آن هستند .

۲-۲ آشنایی مختصر با زبان OWL

همانطور که می‌دانید ما تا به حال ۳ نسل از وب را پشت سر گذاشته‌ایم:

۱. وب ۱ یا وب سنتی همان وبی که آقای «تیم برنرزلی» اختراع کرد و حاوی یک سری صفحات ساده که در آن‌ها فقط متن و عکس بود.

۲. وب ۲ یا اشتراکی : وبی که ظهور شبکه‌های اجتماعی (Social-Networks) را شاهد بودیم و کاربران امکان اشتراک هر نوع داده را داشتند.

۳. وب ۳ یا وب معنایی (Semantic-Web) : در وب ۳ صفحات به طور کامل برای موتورهای جستجو و هر نوع موتور تحلیل‌گر دیگر باید شناخته شده باشد و متون برای آن فقط یک سری

کلمه‌ی ساده در نظر گرفته نشود.

روش‌های معنا بخشیدن به صفحات وب :

۱. XML

۲. URI

۳. RDF ، OWL

زبان هستی شناسی وب یا همان OWL یک زبان وب معنایی می باشد که برای بیان مفاهیم و دانش پیچیده در رابطه با اشیا موجودیت ها گروه‌های اشیا و روابط بین آنها ایجاد شده است OWL یک زبان محاسباتی منطقی می باشد به نحوی که دانشی که توسط OWL بیان می گردد می‌تواند توسط برنامه های کامپیوتری مورد استفاده قرار گیرد. برای مثال برای بررسی کردن سازگاری و پایداری یک ساختار دانش و یا آشکار کردن دانش ضمنی کاربرد دارد. اسناد OWL که آن را آنتولوژی یا هستان شناسی می نامیم می تواند در فضای وب منتشر شود می تواند به سایر هستیان شناسی های OWL ارجاع دهد و یا توسط آنها مورد ارجاع قرار گیرد.OWL بخشی از مجموعه تکنولوژی وب معنایی می باشد که شامل RDF، RDFS، وSPARQL می باشد. یکی از ابزار های مرتبط با هستی شناسی که در این پروژه از آن استفاده شده است protégé میباشد که از مزایای آن موارد زیر می باشد:

- طراحی و ساخت آنتولوژی های owl بدون نیاز به کدنویسی
- دارای موتور استنتاج جهت ارزیابی سازگاری هستان شناسی (آنتولوژی)
- خروجی گرافیکی از ساختار هستان شناسی
- قابلیت تعریف انواع ویژگی ها برای آنتولوژی

۲-۲-۱ تفاوت آنتولوژی با گراف دانش

از آنجایی که برنامه های معنایی به سرعت در حال تبدیل به موضوعات مهم در صنعت می شوند، سوالاتی غالباً در رابطه با آنتولوژی ها و گراف های معنایی پیش می آید به خصوص در رابطه با تفاوت این دو.

آیا آنتولوژی ها و گراف های دانش یکسان هستند؟ اگر نیستند تفاوتشان چیست؟ ارتباط بین این دو چیست؟ آنتولوژی ها مدل‌های داده معنایی هستند که نوع اشیایی که در حوزه مورد نظر ما وجود دارند و ویژگی هایی که می‌توانند برای توصیف آنها مورد استفاده قرار بگیرند را تعریف می‌کنند. در واقع می‌توان گفت آنتولوژی ها مدل‌های داده عمومی یا کلی هستند و فقط به توصیف کلی یا عمومی چیزهایی که ویژگیهای معینی دارند می‌پردازد، اما به اطلاعات مشخص در رابطه با اعضا و موجودیت‌های حاضر در آن حوزه نمی پردازد. از سه بخش اصلی تشکیل شده است که به صورت زیر توصیف می‌شود:

- کلاس ها: یعنی انواعی از چیزها که دارای مشخصات معین و منحصربه فرد هستند که در داده های ما وجود دارد
- ارتباطات: ویژگی هایی که دو کلاس را به یکدیگر مرتبط می کنند
- خصیصه‌ها: ویژگی هایی که کلاس یک عضو را مشخص می کنند

پس هستی شناسی به مشخصات رسمی یک مفهوم سازی اشاره دارد، یعنی نمایشی ساختاریافته و استاندارد شده از مفاهیم و روابط در یک حوزه خاص که معمولاً شامل سلسله مراتبی از مفاهیم، مجموعه ای از روابط بین آن مفاهیم، و مجموعه ای از بدیهیات یا قوانینی هستند که بر رفتار آن مفاهیم و روابط حاکم هستند ولی گراف دانش یک پایگاه داده یا ساختار داده ای است که دانش را به عنوان مجموعه ای از موجودیت های به هم پیوسته و روابط آنها با یکدیگر نشان می دهد و یک پیاده سازی مشخص از مشخصات هستی شناسی در یک پایگاه داده یا ساختار داده است. هستی شناسی ها برای تعریف ساختار و معناشناسی یک گراف دانش استفاده می شوند، در حالی که نمودارهای دانش برای ذخیره و پرس و جو از داده های ارائه شده توسط هستی شناسی استفاده می شوند. یک گراف دانش هنگامی ایجاد می شود که شما یک آنتولوژی (همان دیتا مدل) را به یک دیتاست از داده های معین اعمال کنید. به بیان دیگر می توان گفت:

$$\text{Ontology} + \text{Data} = \text{Knowledge Graph}$$

اما به طور کل هستی شناسی ها و نمودارهای دانش مفاهیم مکملی هستند که اغلب با هم در برنامه هایی مانند یکپارچه سازی داده ها، پردازش زبان طبیعی و هوش مصنوعی استفاده می شوند.

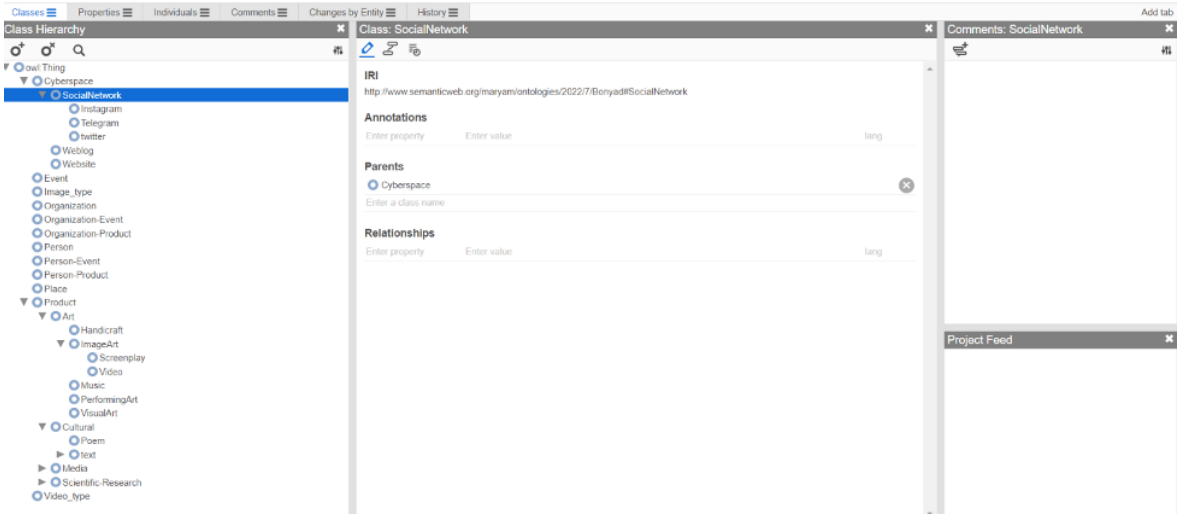
فصل ۳

روش پیشنهادی برای انجام پروژه

۳-۱ جمع آوری داده

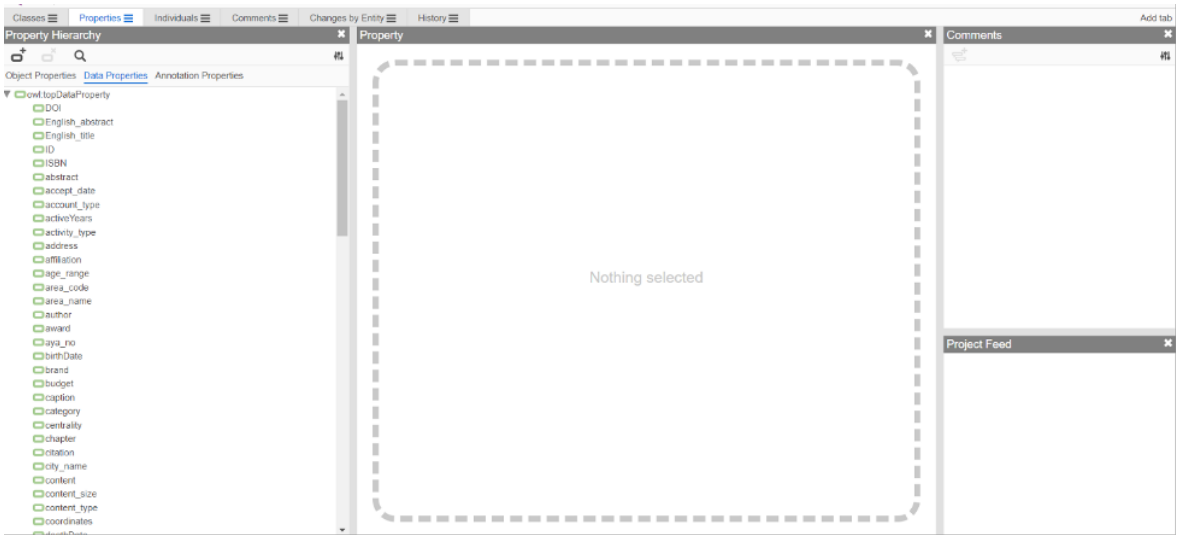
در انجام این پژوهش داده های جمع آوری شده در قالب فایل OWL ارائه شده است که همانطور که گفته شد OWL نوعی فرمت فایل است که در زمینه فناوری‌های وب معنایی برای نمایش هستی‌شناسی‌ها، که توصیف‌های رسمی حوزه‌های دانش هستند، استفاده می‌شود که معمولاً به عنوان سلسله مراتبی از مفاهیم و روابط آنها نشان داده می شوند که می توانند برای توصیف ساختار یک حوزه خاص و روابط بین عناصر مختلف آن استفاده شوند. در زمینه Neo4j، یک فایل OWL می تواند منبع اطلاعاتی ارزشمندی برای ایجاد یک پایگاه داده گراف باشد. با استخراج داده ها از یک فایل OWL و استفاده از آن برای ایجاد گره ها و روابط در یک پایگاه داده گراف، می توان یک مدل داده قدرتمند و منعطف ایجاد کرد که به طور دقیق ساختار یک حوزه دانش خاص را نشان می دهد. این رویکرد امکان پرس و جو و تجزیه و تحلیل کارآمد داده های پیچیده و بهم پیوسته را فراهم می کند و آن را برای برنامه هایی که با حجم زیادی از داده ها از منابع متعدد سروکار دارند ایده آل می کند. به طور کلی، استفاده از یک فایل OWL برای ایجاد یک پایگاه داده گراف در Neo4j می‌تواند مزایای متعددی از جمله مدل‌سازی و تحلیل داده‌ها کارآمدتر، کیفیت بهتر داده‌ها و انعطاف‌پذیری بیشتر در انطباق با نیازهای متغیر داده در طول زمان را ارائه دهد. Neo4j با توانایی مدیریت مجموعه داده‌های پیچیده و به هم پیوسته، ابزاری قدرتمند برای مدیریت و تجزیه و تحلیل داده‌ها از طیف گسترده‌ای از منابع است و استفاده از فایل‌های OWL می‌تواند به باز کردن بینش‌های بیشتر از این مجموعه داده‌ها کمک کند. در تصویر زیر به کمک

نرم افزار protégé محتویات فایل را مشاهده میکنیم تا از آن مفاهیم مورد نیاز را استخراج نماییم:



شکل ۳-۱

همانطور که مشاهده میکنید سلسله مراتبی از کلاس ها را داریم که هر کدام دارای زیرشاخه و روابط با یکدیگر هستند.



شکل ۳-۲

در این قسمت سلسله مراتب ویژگی ها را به صورت مجزا داریم که با بررسی هرکدام، ویژگی هر موجودیت یا کلاس را استخراج کرده و در گراف دانش‌مان به عنوان خصیصه های گره اضافه کردیم. سپس روابط بین کلاس ها را بر اساس Domain و Range که گره شروع و پایان برای آن یال را مشخص میکند را جدا کردیم تا در گراف دانش مان پیاده‌سازی کنیم.

۳-۲ ایجاد گراف دانش

در پایگاه داده های گراف، داده ها در گره ها و روابط ذخیره می شوند. گره ها موجودیت ها یا اشیاء را نشان می دهند، در حالی که روابط نشان دهنده ارتباطات یا تعاملات بین آن موجودیت ها هستند. پس از شناسایی و استخراج موجودیت ها و روابط، موجودیت ها، هر موجودیت یا کلاس در فایل OWL باید با یک گره در گراف دانش مطابقت داشته باشد و هر رابطه باید با یک یال بین گره ها ترسیم شود. پس ویژگی ها را به گره ها و یال ها در نمودار دانش اختصاص دادیم. نمودار دانش به دست آمده همان اطلاعات فایل OWL را نشان خواهد داد اما در قالبی ساختاریافته و قابل خواندن توسط ماشین. هر گره می تواند یک یا چند برچسب برای دسته بندی نوع موجودیتی که نشان می دهد داشته باشد. برچسب‌ها در اصل برچسب‌هایی هستند که به ما کمک می‌کنند داده‌های خود را سازماندهی کرده و پرس و جو را آسان‌تر کنند. برای مثال، برچسب‌هایی مانند «شخص»، «محصول» یا «مکان» داریم. علاوه بر برچسب‌ها، گره‌ها همچنین می‌توانند یک یا چند ویژگی داشته باشند که جفت‌های کلید_مقدار هستند که اطلاعات مربوط به گره را ذخیره می‌کنند. به عنوان مثال، یک گره "Person" دارای ویژگی هایی مانند name، age، email است. روابط گره ها را به هم متصل می کنند و همچنین می توانند برچسب هایی برای توصیف نوع تعامل بین آنها داشته باشند. برای مثال، برچسب‌های رابطه‌ای مانند knows has_location، ، has_Participant داریم. هنگامی که پایگاه داده گراف را پرس و جو می کنید، می توانید از برچسب ها، ویژگی ها و روابط برای یافتن و تجزیه و تحلیل زیرمجموعه های خاصی از داده ها استفاده کنید.

۳-۳ معرفی Neo4j

Neo4j یک سیستم مدیریت پایگاه داده گراف قدرتمند و همه کاره است که در سال های اخیر به دلیل توانایی آن در مدیریت ساختارهای داده پیچیده و به هم پیوسته محبوبیت پیدا کرده است. Neo4j با تاکید بر روابط بین نقاط داده و توانایی آن برای انجام پرس و جوهای بسیار کارآمد در مجموعه داده های بزرگ، به گزینه ای برای توسعه دهندگان و تحلیلگران داده در صنایع مختلف تبدیل شده است. برخلاف پایگاه‌های داده رابطه‌ای سنتی، که داده‌ها را در جداول دارای ردیف و ستون ذخیره می‌کنند، Neo4j داده‌ها را به عنوان مجموعه‌ای از گره‌ها و یال‌ها نشان می‌دهد، جایی که گره‌ها موجودیت‌ها و یال‌ها

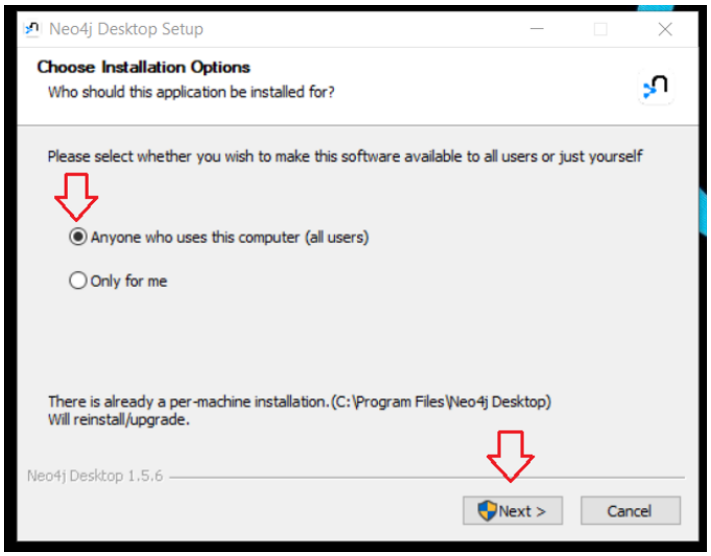
روابط بین آن موجودیت‌ها را نشان می‌دهند. این رویکرد امکان ایجاد مدل‌های داده بسیار انعطاف‌پذیر را فراهم می‌کند که می‌توانند به راحتی در صورت نیاز اصلاح و به روز شوند، و آن را برای برنامه‌هایی که با داده‌های بسیار بهم پیوسته سروکار دارند ایده‌آل می‌کند.

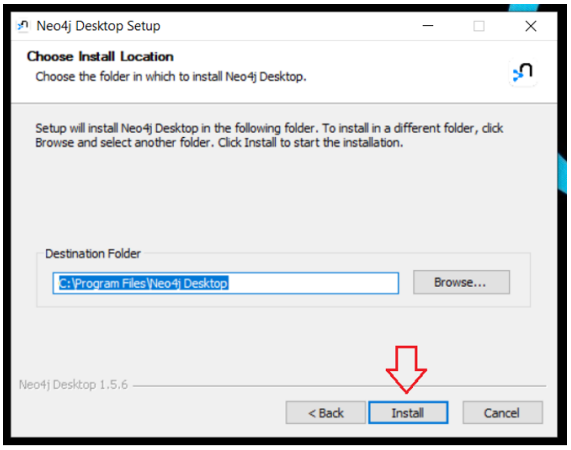
۳-۳-۱ نصب و راه‌اندازی Neo4j

ابتدا به نصب و راه اندازی نسخه ۶.۵.۱ آن میپردازیم: به سایت www.neo4j.com مراجعه کرده و از قسمت neo4j.com/download وارد بخش Started Get میشویم که باید ابتدا فرم مشخصات را پر کنیم و موافقت نامه مجوز Neo4j را برای نرم افزار دسکتاپ Neo4j اعلام کنیم تا فایل مورد نظر شروع به دانلود شود. سیستم مورد نیاز توصیه شده برای نصب

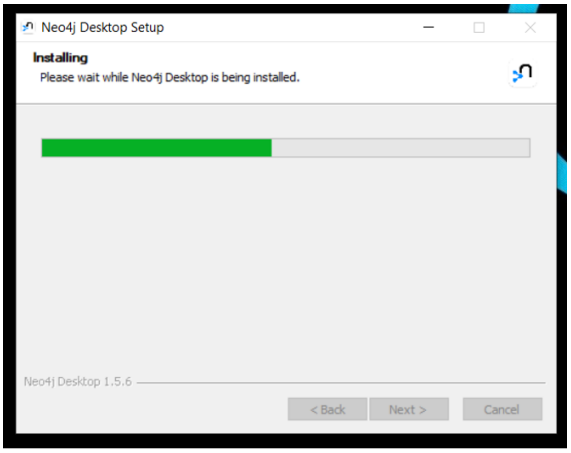
MacOS 10.10 (Yosemite)+, Windows 8.1+ with Powershell 5.0+, Ubuntu 12.04+, Fedora 21, Debian 8

تعیین شده است. در صفحه بعد Key Activation تهیه شده است که باید از این کلید برای فعال کردن کپی DesktopNeo4j استفاده کنید. بعد از دانلود شدن و اجرای installer طبق تصاویر زیر پیش می‌رویم:

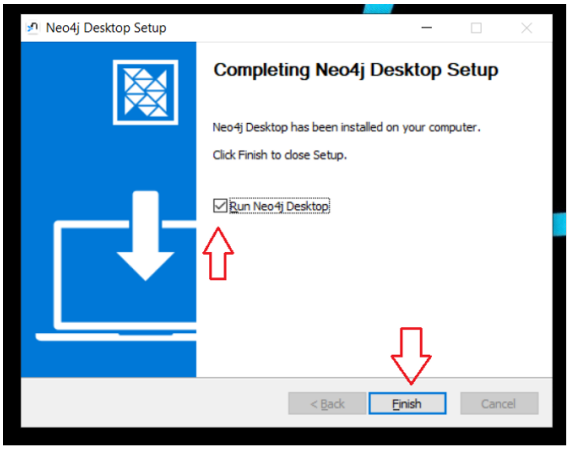




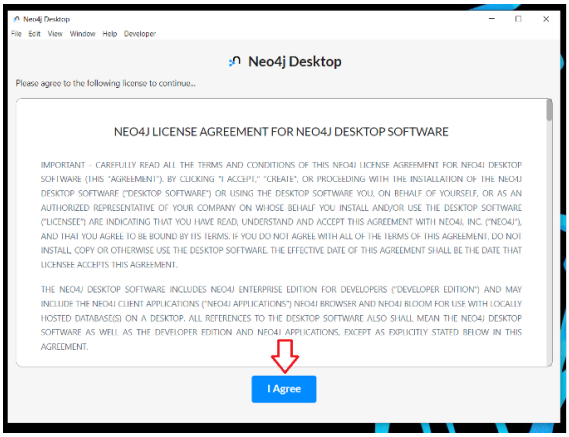
شکل ۳-۳



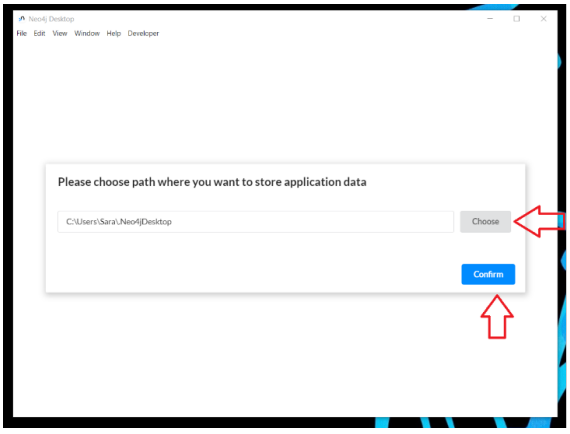
شکل ۳-۴



شکل ۳-۵

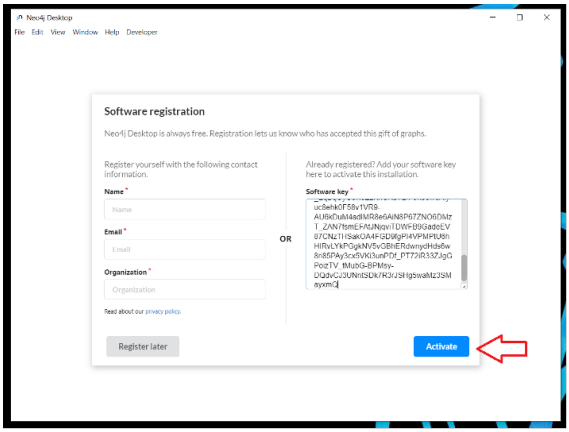


شکل ۳-۶

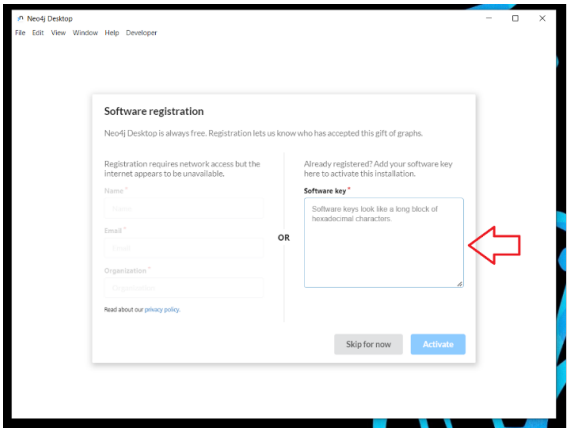


شکل ۳-۷

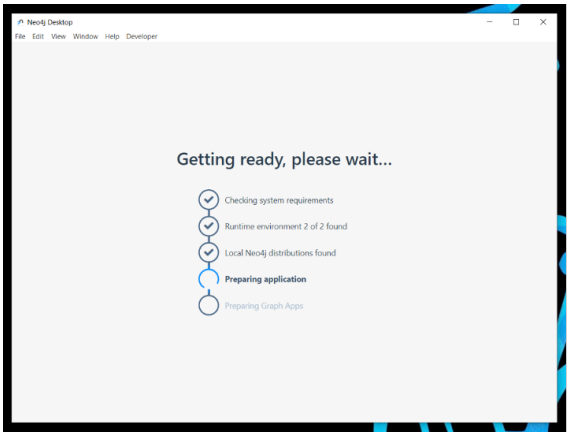
در این قسمت Key Activation داده شده در قبل را وارد کنید. و یا می‌توانید با پر کردن فرم سمت راست صفحه برنامه، کلیدی را از داخل برنامه ایجاد کنید.



شکل ۳-۸

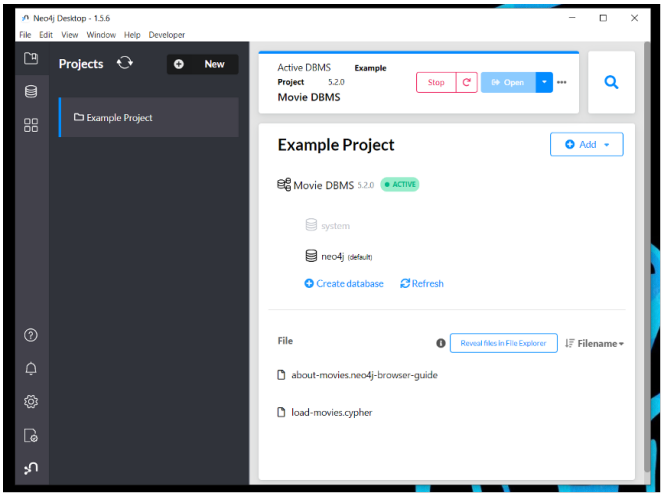


شکل ۳-۹



شکل ۳-۱۰

برنامه با موفقیت نصب شده و صفحه پنل کاربری زیر نمایان میشود:



شکل ۳-۱۱

۳-۳-۲ تعامل با رابط کاربری Neo4j

رابط کاربری Neo4j دارای سه بخش زیر است:

۱. Sidebar

۲. Editor

۳. Stream

برای ایجاد پایگاه داده، در سمت راست بر روی دکمه Add و Graph New کلیک کرده گزینه graph local a Create را انتخاب کنید سپس نام دیتابیس مورد نظر و پسوورد را وارد کرده و روی Create کلیک نمایید. پس از ایجاد پایگاه داده، روی دکمه Start کلیک کنید. Open را که بزنید صفحه BrowserNeo4j باز میشود. مرورگر Neo4j یک رابط بصری غنی برای تعامل با پایگاه داده گراف فراهم می کند. این شامل یک پنل در سمت چپ صفحه است که به شما امکان می دهد طرحواره پایگاه داده گراف را بررسی کنید، پرس و جوها را اجرا کنید و تنظیمات پایگاه داده را مدیریت کنید. این ویرایشگر به شما اجازه می دهد تا کوئری های Cypher را بنویسید و اجرا کنید. راه دیگر برای داشتن مرورگر Neo4jاین است که آدرس http://127.0.0.1:7474 را درمرورگر خود وارد کنید و سپس وارد ویرایشگر کوئری شوید. برای بار اول، از شما نام کاربری و رمز عبوری را درخواست می‌کند که همان نام و رمز عبور، هنگام نصب است.

مرورگر یک shell command Cypher تعاملی است که به شما امکان می دهد با گراف خود تعامل داشته باشید و اطلاعات موجود در آن را visualize کنید . حال کمی با زبان cypher آشنا شویم :

برای ساخت یک آبجکت از CREATE استفاده میکنیم که به فرمت زیر است:

```
CREATE (VariableName:objectType (property1 :value1 ,property2 : value2 ,...  })
```

این معادل ساخت جدول در بانک‌های اطلاعاتی رابطه‌ای است.

برای بازیابی یک آبجکت، از کوئری زیر استفاده می‌شود:

```
MATCH (VariableName:objectType ) RETURN VariableName
```

که میتوان با where برای آن شرط تعیین شود .

```
MATCH (n:Person)-[:KNOWS]->(m:Person)
WHERE n.name = 'Alice'
RETURN m AS person
```

برای یافتن و بازنمایی همه گره ها از عبارت های زیر استفاده میکنیم :

```
MATCH (n) RETURN n
```

برای ایجاد رابطه بین شی a و b که شروع از a و به سمت گره b است به شکل زیر کوئری مینویسیم که نوع رابطه را داخل براکت مشخص میکنیم :

```
CREATE (a)-[:relationship_type]->(b)
```

برای update یا ایجاد یک property از set استفاده میشود:

```
SET e.property1 = $value1
```

برای حذف رابطه با مشخصات داده شده:

```
MATCH (n:Label)-[r]->(m:Label)
WHERE r.id = 123
DELETE r
```

طبیعی است گستردگی این زبان به این موارد محدود نمیشود برای اطلاعات کامل‌تر به داکیومنت خود سایت Neo4j میتوان مراجعه کرد.

فصل ۴

نتایج بدست آمده

۴-۱ پیاده‌سازی طرح کلی گراف دانش

پس از راه اندازی Neo4jو جمع آوری و استخراج داده ها آغاز به ایجاد گراف دانش مان میکنیم. پایگاه داده گراف‌مان از گره ها و یال ها تشکیل شده است که در آن گره ها موجودیت هایی مانند افراد، سازمان ها، محصولات و رویدادها را نشان می دهند و یال ها روابط بین این موجودیت ها را نشان می دهند. ابتدا از نوع کلاس thing یک کلاس Cyberspace میسازیم. سپس سه گره با برچسب های «وبلاگ» و «وب سایت» و یک گره دیگر با برچسب «شبکه اجتماعی» ایجاد می کنیم که زیرکلاس های فضای مجازی هستند. این گره ها با استفاده از رابطه "IS_A" به گره Cyberspace متصل می‌کنیم.

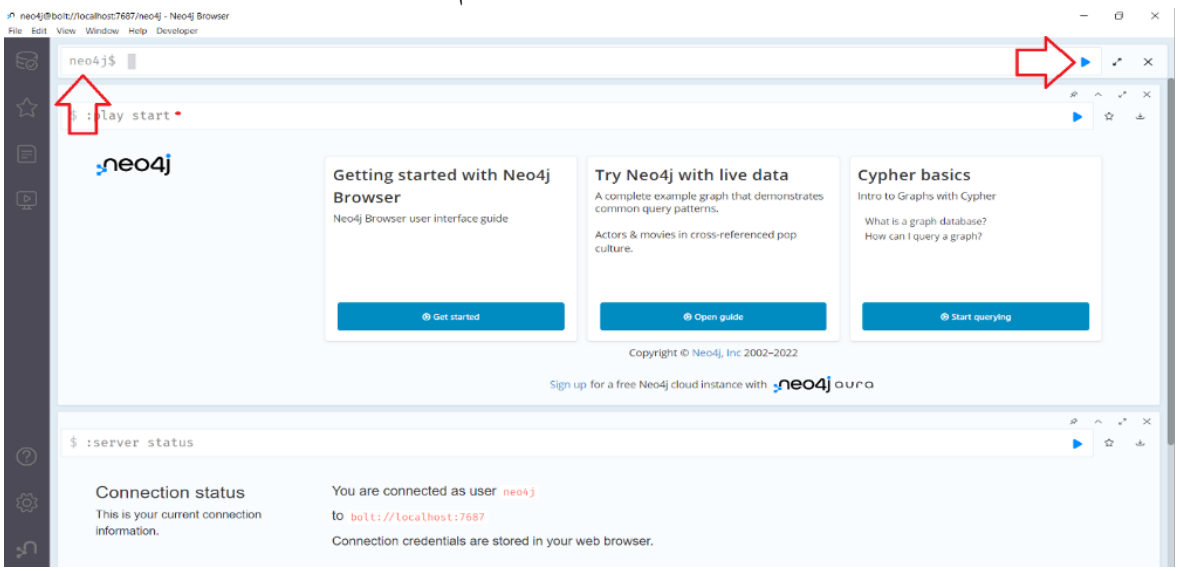
سپس به ساخت زیرکلاس های Social-Network میپردازیم که Instagram و Telegram و Twitter هستند. از آنجایی که برای property های تعریف شده مقادیری در فایل owl داده نشده است و مقدار خالی در Neo4jبه منزله عدم وجود آن ویژگی است، از مقدار ۰ برای مقادیر عددی و از NULL برای رشته استفاده کردیم که بعدا طبق داده های بدست آمده کامل، آن ها را آپدیت کنیم.

```
CREATE (Cyberspace:thing {title: "Cyberspace"}),
       (Weblog:Cyberspace {title: "Weblog", url:"NULL", description:"NULL", person_name:"NULL"}),
       (Website:Cyberspace {title: "Website", url:"NULL", description:"NULL", affiliation:0}),
       (SocialNetwork:Cyberspace {title: "SocialNetwork", account_type:"NULL", description:"NULL"})
CREATE (Weblog)-[:is_a]->(Cyberspace),
       (Website)-[:is_a]->(Cyberspace),
       (SocialNetwork)-[:is_a]->(Cyberspace)

CREATE (Instagram:SocialNetwork {title:"Instagram",  InstagramID:"NULL"})-[:is_a]->(SocialNetwork),
       (Telegram:SocialNetwork {title:"Telegram", TelegramID:"NULL"})-[:is_a]->(SocialNetwork),
       (Twitter:SocialNetwork {title:"Twitter", TwitterID:"NULL"})-[:is_a]->(SocialNetwork)
```

شکل ۴-۱

سپس در این قسمت از browser کوئری هارا وارد میکنیم:

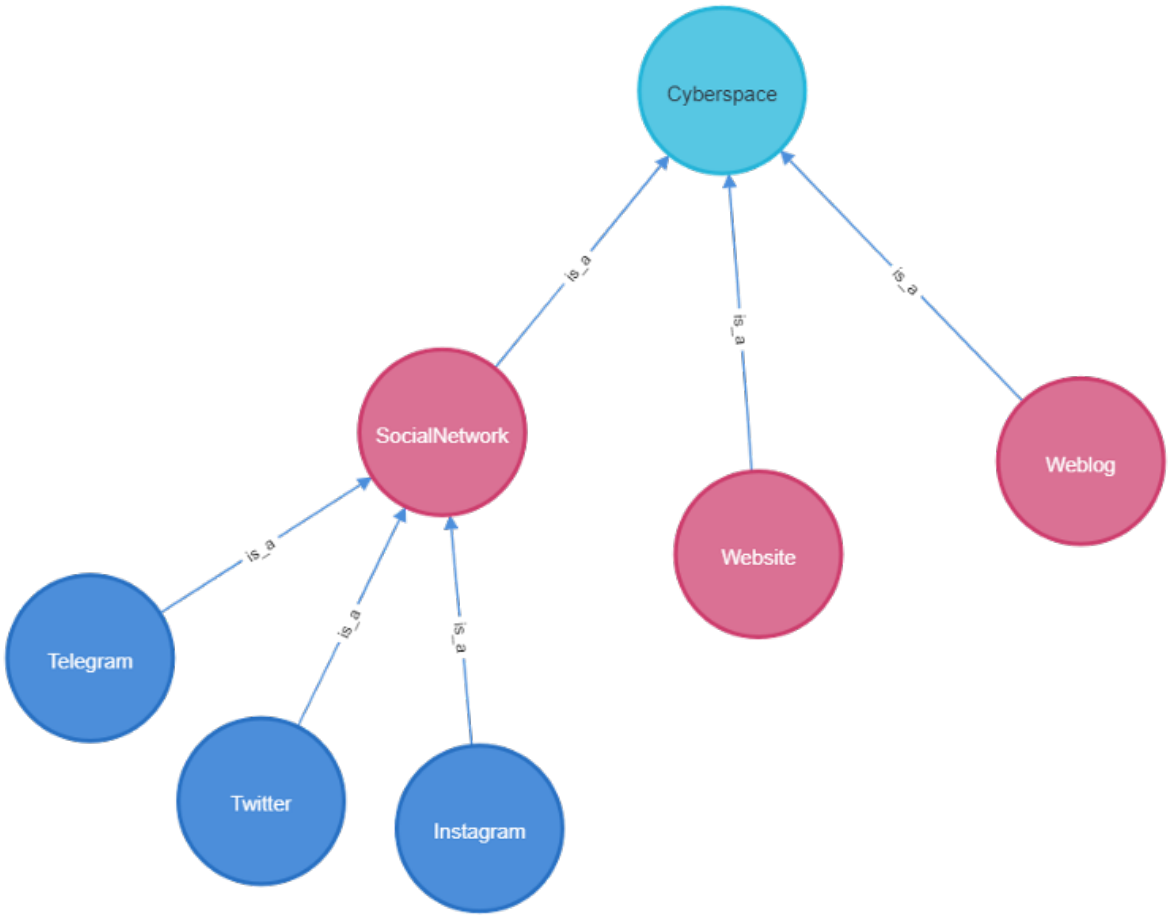


شکل ۴-۲

که برای کوئری بالا خروجی زیر را میدهد و گره ها و روابط را ایجاد میکند:

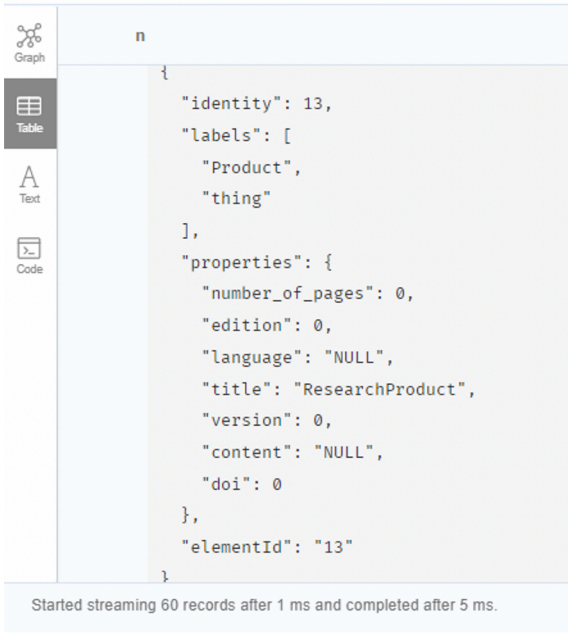
شکل ۴-۳

با نوشتن کوئری $MATCH(n)RETURNn$ کل آنچه تا کنون ساخته‌ایم را میتوانیم به صورت گراف مشاهده کنیم که خروجی کوئری بالا به شکل زیر است:



شکل ۴-۴

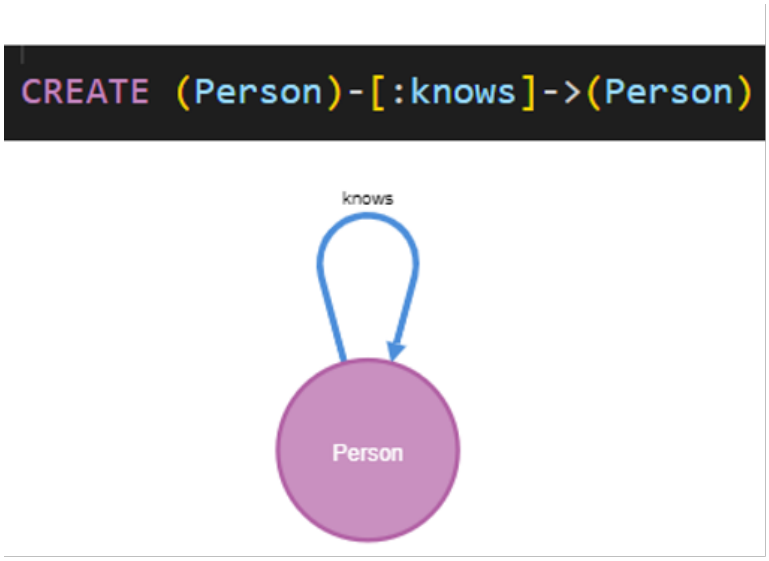
در قسمت view table میتوانیم property های هر گره را مشاهده کنیم:



شکل ۴-۵

سپس، گره های دارای برچسب های Person، Event، Organization، Place Product و Person_Product را ایجاد کردیم. سپس چندین زیرگروه از این گره ها را ایجاد و با استفاده از رابطه "IS_A" آنها را به هم متصل نمودیم.

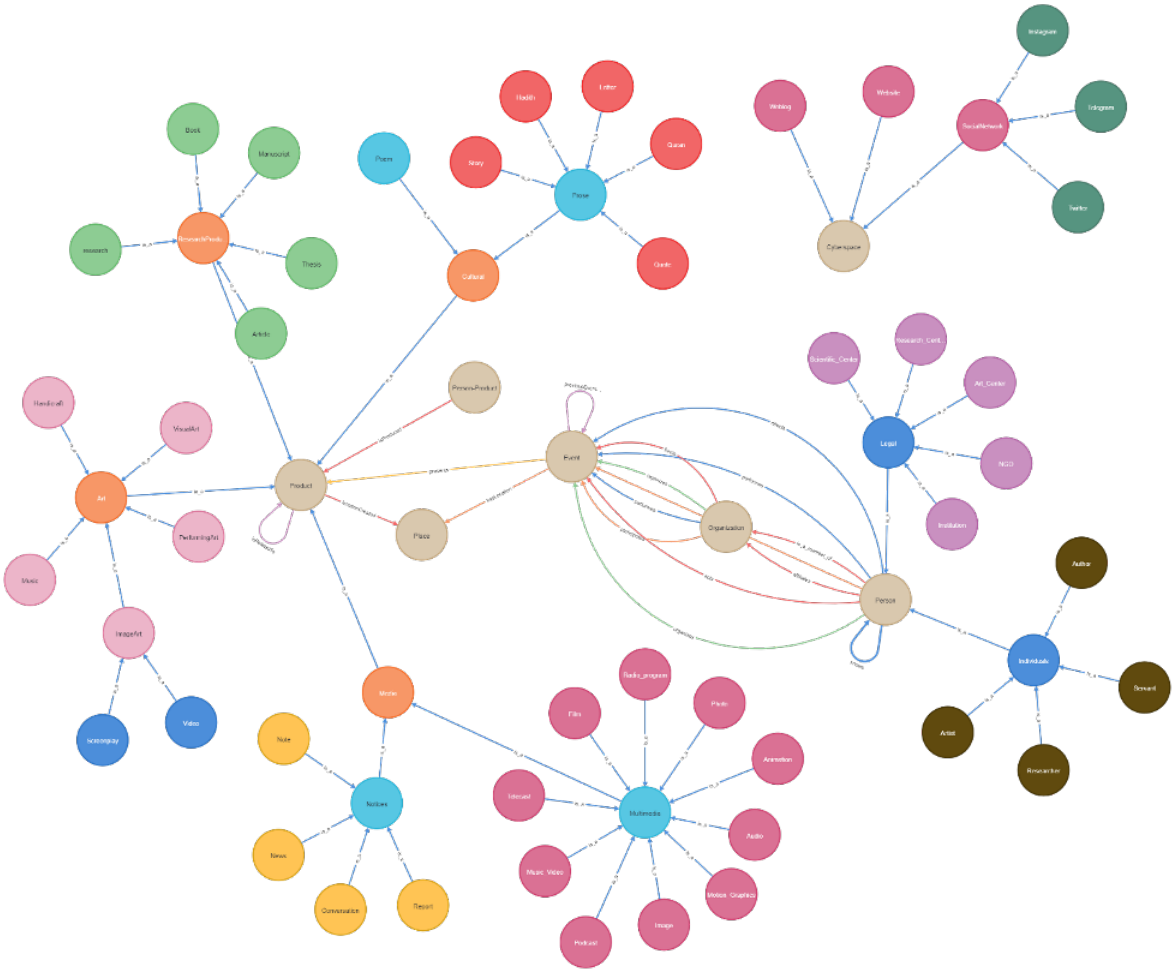
پس از ایجاد گره ها و زیرگروه ها، روابط و لبه هایی را بین آنها ایجاد کردیم تا روابط غیر زیرکلاس را نشان دهیم. به عنوان مثال، لبه ای ایجاد کردیم تا نشان دهد که یک فرد شخص دیگری را می شناسد یعنی یک self-relation به خود کلاس Person ، که این امکان در Neo4jساپورت میشود و به صورت زیر است:



شکل ۴-۶

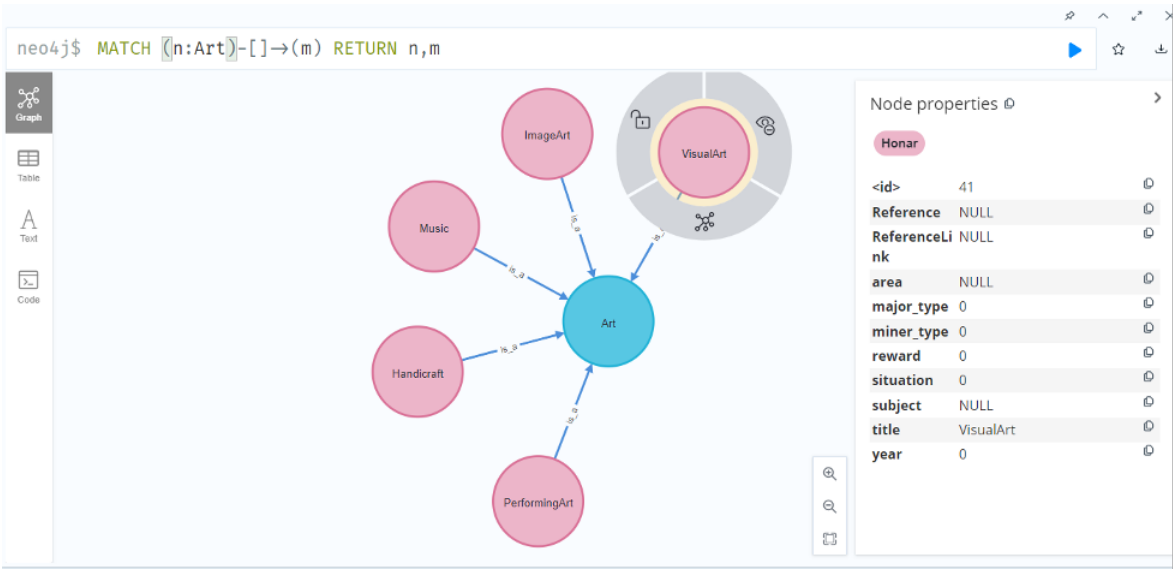
از روابط دیگری که استخراج شد یک نمونه این هم بود که انسان یک رویداد را سازماندهی می کند، در رویداد اقدام می کند، در رویداد شرکت می کند، رویداد را اجرا می کند، رویداد را هدایت می کند، به یک سازمان وابسته است، و عضو یک سازمان است. پس از آن همچنین لبه هایی ایجاد شد تا نشان دهد که سازمان بودجه رویداد را تامین می کند، رویداد را سازماندهی می کند، رویداد را اجرا می کند و در رویداد شرکت می‌کند.

برای رویداد اضافه کردیم که رویداد دارای مکان است، محصول را ارائه می دهد و به رویداد دیگری مرتبط است. لبه های دیگری نیز ایجاد شد تا نشان دهد که یک محصول فرهنگی یک نوع فرعی از محصول است و برای مثال یک شعر و یک نثر زیرگونه های محصول فرهنگی هستند. در نهایت بعد از ساخت همه گره ها و روابط شان که حدود ۹۱ گره و ۳۴۱ ویژگی و ۷۴ رابطه شد، نمای کلی گرافی که بدست آمد به شکل زیر است:



شکل ۴-۷

در گراف دانش ساخته شده، با کمك پرس و جوهاي متعدد مي توان اطلاعات متفاوتی استخراج کرد. مثلا اگر بخواهیم لیستی از تمام هنر هایی که داریم مجزا مشاهده کنیم کوئری زیر را نوشته و گره های زیرمجموعه بدست می‌آید که با کلیک روی هر گره در سمت راست قسمت properties Node ویژگی های آن گره را به اختصاص نشان میدهد.



شکل ۴-۸

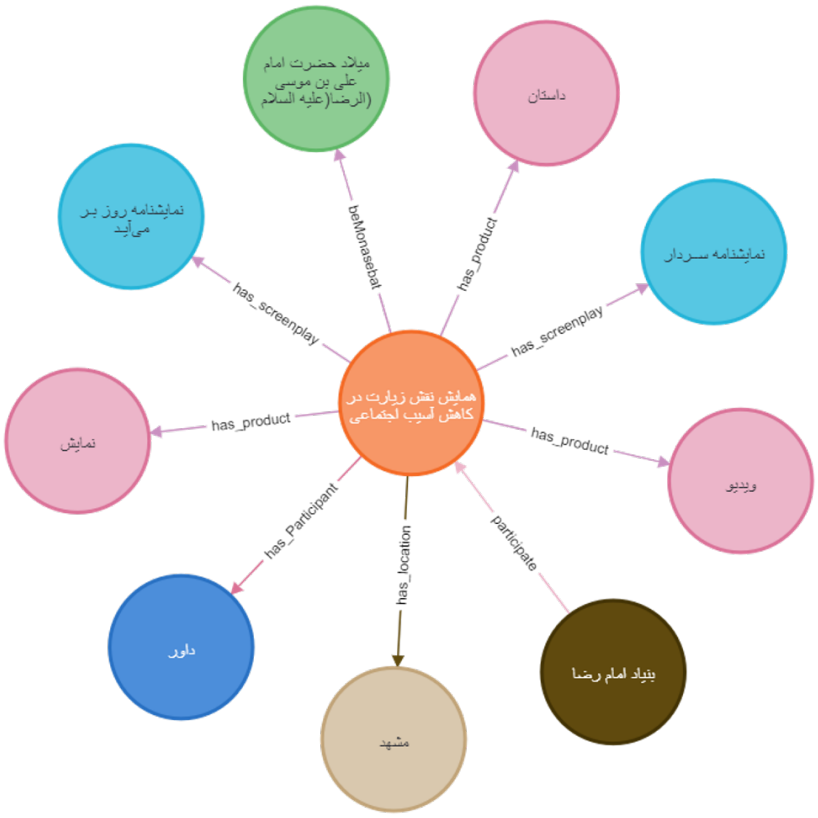
۴-۲ گراف دانش پروژه بنیاد

بعد از ایجاد این طرح کلی، از سایت بنیاد امام رضا (ع) جزییات بیشتری راجع به جشنواره ها استخراج کردیم و این گراف را با داده های این بنیاد بازنمایی نمودیم . ۱۲ جشنواره داریم اعم از یادمان شهری رضوی، همایش ایثار و شهادت، همایش نقش زیارت در کاهش آسیب اجتماعی و... که در شهر های مشهد و سبزوار به مناسبت میلاد حضرت امام علی بن موسی الرضا(علیه السلام) توسط این بنیاد برگزار شده است. این داده ها را وارد پایگاه داده گرافی مان کرده و گراف دانش زیر را بدست آوردیم:

شکل ۴-۹

برای نمونه برای آگاهی از جشنواره انتخاب کتاب سال رضوی تا ببینیم چه محصولاتی ارائه کرده در کجا برگزار میشود و توسط چه سازمانی، کوئری زیر را وارد نموده و اطلاعات آن را بدست می‌آوریم:

```
MATCH (e:Event {Ev:1})-[r]-(n)
RETURN e, r, n
```

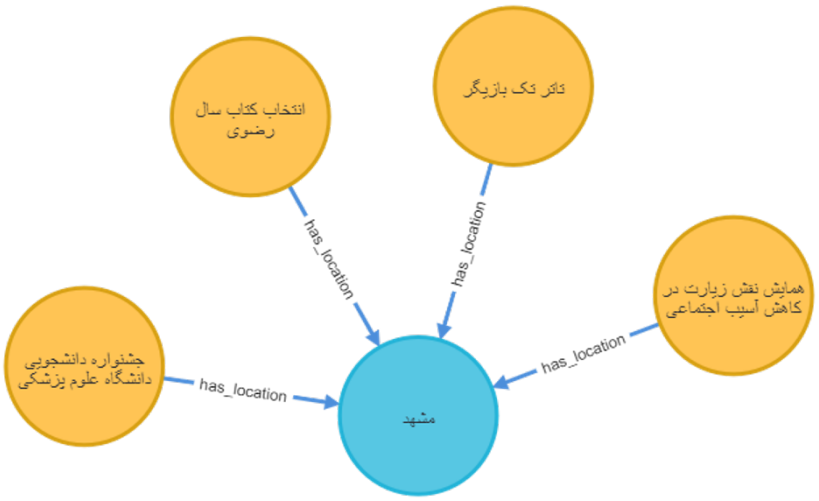


شکل ۴-۱۰

یا مثلا اگر بخواهیم ببینیم فقط در شهر مشهد چه همایش هایی برگزار می‌شود کوئری زیر را وارد کرده و جشنواره ها را مشاهده میکنیم:

```
MATCH (m)-[]->(n:Place {city_name:"مشهد"}) RETURN n , m
```

شکل ۴-۱۱

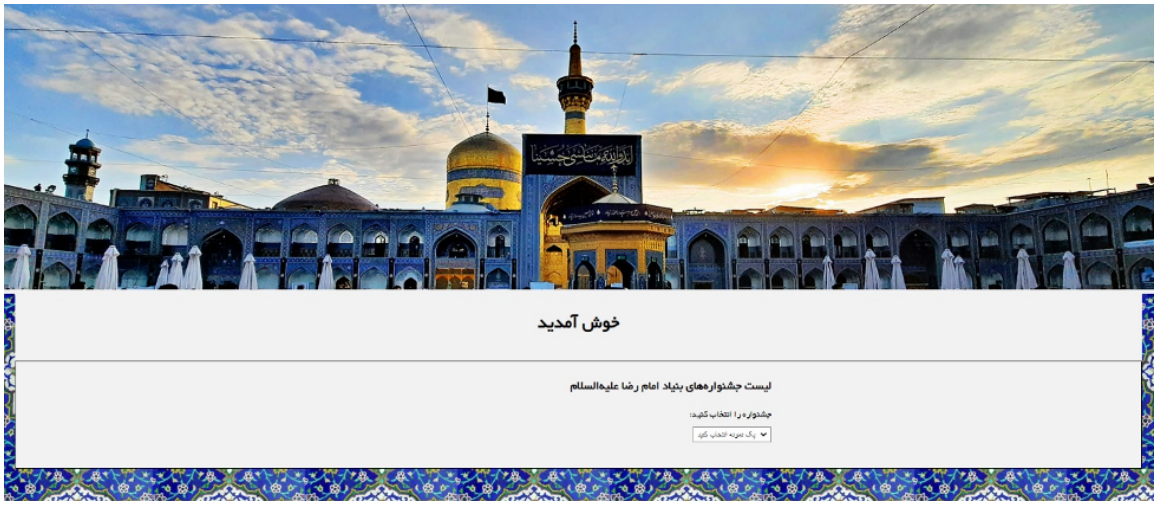


شکل ۴-۱۲

۴-۳ رابط کاربری

جهت داشتن یک رابط کاربری مناسب که کاربر با کوئری های پیچیده درگیر نشود و به آسانی اطلاعات این جشنواره ها را دریافت نماید نیز یک صفحه وب طراحی کرده‌ایم تا با انتخاب گزینه مراسم ها در بک گراند کوئری ها را نوشته و از پایگاه داده گرافی مان پاسخ پرس و جوی مورد نظر را برای کاربر نشان دهیم که نمونه هایی از آن به شکل زیر است:

صفحه اصلی:



شکل ۴-۱۳

انتخاب جشنواره پنجم توسط کاربر از قسمت گزینه ها و نمایش گراف نتیجه:

شکل ۴-۱۴

فصل ۵

جمع بندی و کار های پیشرو

این پایان نامه کارشناسی استفاده از Neo4j را برای ایجاد یک گراف دانش نشان داده است. از طریق اجرای این فناوری، مشاهده شد که چگونه می توان از آن برای سازماندهی و ساختار داده ها به گونه ای استفاده کرد که تجزیه و تحلیل و تصمیم گیری مؤثرتر را تسهیل کند.

این پژوهش به بررسی مفاهیم اساسی گراف های دانش و زبان هستی شناسی وب پرداخته است. همچنین در مورد مزایای استفاده از Neo4j توضیح داده شد که بعنوان یک سیستم مدیریت پایگاه داده قدرتمند و همه منظوره برای ذخیره، مدیریت و کوئری گراف های در مقیاس بزرگ استفاده می شود. به طور کلی، این پایان نامه یک نمای کلی از فرآیند ایجاد یک گراف دانش با استفاده از Neo4j، از چارچوب مفهومی تا جزئیات پیاده سازی ارائه کرده است.

در طول فرآیند انجام پروژه، ما با چالش های مختلفی مانند شناسایی منابع داده مربوطه، طراحی یک طرحواره مناسب و بهینه سازی عملکرد کوئری ها و عدم وجود راهنمای کاربری Neo4j مواجه بوده ایم. با این حال، ما استراتژی هایی را برای مقابله با این چالش ها ایجاد کرده ایم، مانند استفاده از تکنیک های پیش پردازش داده ها و تحلیل هستی شناسی های موجود.

در این پژوهش گراف دانش مورد نظر براي داده‌های بنیاد امام رضا (ع) ساخته شد و برای آن رابط کاربری جهت سهولت استفاده ایجاد گردید. نتایج پیاده سازی های ما بر روی این داده ها نشان داده است که ایجاد یک گراف دانش با استفاده از Neo4j می‌تواند منجر به بهبود یافته‌ها، کشف دانش و تصمیم‌گیری بهتر شود. مطالعات موردی ما نشان داده است که Neo4j یک انتخاب مناسب برای

برنامه‌هایی است که نیاز به تجزیه و تحلیل بلادرنگ و پردازش داده‌های پیچیده و به هم پیوسته دارند، و آن را به یک انتخاب محبوب برای موارد استفاده این‌چنینی تبدیل می‌کند.

ما امیدواریم که این تحقیق به مجموعه دانش رو به رشد در این زمینه کمک کند و الهام بخش کار آینده در مدیریت و تجزیه و تحلیل داده های مبتنی بر گراف باشد.

با افزایش داده و به روزرسانی مقادیر گراف دانش، طبیعی است اطلاعات متنوع و سودمند تری به‌دست خواهیم آورد، که یکی از راه های جمع آوری دیتا استفاده از خزشگر های وب است که حاکی از راه طولانی مسیر پیش رو میباشد و نشان می‌دهد کار این پروژه در اینجا به پایان نمی‌رسد.

بعنوان کار پیشرو ادغام با فناوری‌های یادگیری ماشین و هوش مصنوعی می توان نام برد. توانایی Neo4j در مدل‌سازی روابط پیچیده، آن را به انتخابی ایده‌آل برای یادگیری ماشین و کاربردهای هوش مصنوعی تبدیل می‌کند. کار آینده می تواند بر روی ادغام Neo4j با چارچوب های یادگیری ماشین تمرکز کند تا زمینه ساز تجزیه و تحلیل ها و پیش بینی های پیشرفته تر باشد.

مراجع

1. neo4j.com
2. https://neo4j.com/docs/cypher-manual/current/
3. neo4j.com/developer/cypher/guide-cypher-basics/
4. www.shamstoos.ir/fa/news/bonyad
5. /www.w3.org/OWL/
6. www.quackit.com/neo4j/tutorial/neo4j_query_language_cypher.cfm
7. documentation.mindsphere.io/MindSphere/howto/howto-create-ontology-owl.html
8. neo4j.com/blog/build-knowledge-graph-from-scratch-even-if-youre-not-full-blown-developer/
9. protege.stanford.edu
10. towardsdatascience.com/how-to-build-a-knowledge-graph-with-neo4j-and-transformers-72b9471d6969
11. farin.academy/google-knowledge-graph/
12. ontology101tutorial.readthedocs.io/en/latest/StartingProtege.html

13. www.javatpoint.com/advantages-of-neo4j

14. https://www.tutorialspoint.com/neo4j/neo4j_data_model.htm

Ferdowsi University of Mashhad

Department of Computer Engineering

B.Sc. Projects

Creating a Knowledge Graph using Neo4J

By:

Sara Asadi

Supervisor:

Dr. Mohsen Kahani

Feb 2023