Ekonomická fakulta
Faculty of Economics

Jihočeská univerzita v Českých Budějovicích
University of South Bohemia in České Budějovice

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Ekonomická fakulta

Katedra aplikované matematiky a informatiky

BAKALÁŘSKÁ PRÁCE

Aplikace metod dataminingu v grafové databázi

Vypracovala: Alexandra Fekete

Vedoucí práce: doc. Ing. Ladislav Beránek, CSc, MBA

České Budějovice 2019

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH
Ekonomická fakulta
Akademický rok: **2017/2018**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Alexandra FEKETE**
Osobní číslo:	**E16534**
Studijní program:	**B6209 Systémové inženýrství a informatika**
Studijní obor:	**Ekonomická informatika**
Název tématu:	**Aplikace metod dataminingu v grafové databázi**
Zadávající katedra:	**Katedra aplikované matematiky a informatiky**

Zásady pro vypracování:

Cílem práce je instalace grafové databáze (např. Neo4j), analýza její činnosti a dále zvládnutí metod data miningu nad touto databází. Následně budou analyzovány a ukázány příklady využití těchto metod na datech získaných z Internetu nebo na jiných reálných datech. Bude předvedeno využití dotazovacího jazyka Cypher. V poslední části bakalářské práce bude provedena analýza vybraných dat (např. ze sociálních sítí, blockchainů, apod.) ve zvolené grafové databázi pomocí vhodných nástrojů (např. Neo4j Serveru a jazyka Cypher) a provedeno zhodnocení této analýzy z hlediska využití.

Metodický postup:

1. Analýza grafové zvolené databáze, její implementace.
2. Analýza, ukázky dotazovacího jazyka nad grafovou databází, jeho popis, analýza metod dataminingu pro grafovou databázi.
3. Extrakce dat ze zvolených zdrojů, výběr cíle analýzy, provedení analýzy v prostředí grafové databáze s využitím nástrojů dataminingu.
4. Závěry, doporučení.

Rozsah grafických prací: **dle potřeby**

Rozsah pracovní zprávy: **40 - 50 stran**

Forma zpracování bakalářské práce: **tištěná**

Seznam odborné literatury:

1. **Introducing the Neo4j Graph Platform The #1 Platform for Connected Data [online]. CA, USA: Neo4j, 2018 [cit. 2018-03-22]. Dostupné z: https://neo4j.com/**
2. **HILLS, Ted. (2016).** *NoSQL and SQL data modeling.* **Basking Ridge, NJ: Technics Publications.**
3. **ROBINSON, Ian, WEBBER James, & EIFREM, Emil. (2013).** *Graph databases.* **Sebastopol, CA: O'Reilly.**
4. **VUKOTIC, Aleksa. (2015).** *Neo4j in action.* **Shelter Island, NY: Manning Publications Co.**

Vedoucí bakalářské práce: **doc. Ing. Ladislav Beránek, CSc.**
Katedra aplikované matematiky a informatiky

Datum zadání bakalářské práce: **19. ledna 2018**

Termín odevzdání bakalářské práce: **12. dubna 2019**

doc. Ing. Ladislav Rolínek, Ph.D.
děkan

JIHOČESKÁ UNIVERZITA
V ČESKÝCH BUDĚJOVICÍCH
EKONOMICKÁ FAKULTA
Studentská 13 (26)
370 05 České Budějovice

doc. RNDr. Jana Klicnarová, Ph.D.
vedoucí katedry

V Českých Budějovicích dne 23. března 2018

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracovala samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47 zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práce k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 10.4. 2019 ……

Alexandra Fekete

Poděkování

Děkuji panu doc. Ing. Ladislavu Beránkovi, CSc., MBA., za cenné připomínky a odborné vedení při tvorbě této bakalářské práce.

Obsah

1 Úvod

S neustálým nárůstem elektronický dat, která je třeba uchovávat a dále s nimi pracovat, vzniká otázka, jakým efektivním a flexibilním způsobem toho dosáhnout. Jednoduchý způsob pro ukládání dat jsou relační databáze, ovšem tento způsob není vždy vyhovující. S vývojem počítačových technologií a narůstajícím množství dat vznikla jiná možnost – NoSQL databáze.

NoSQL databáze přinesly možnost pro data, která například nemají soudržný datový formát či se nedbá na jejich integritu. Data v těchto databázích však kvůli svému množství nejsou nijak modifikována – data neustála přibývají či jsou odstraněny úplně – to může komplikovat způsoby, jak k datům přistupovat. Určitým východiskem je datamining. Pomocí různých metod dataminingu lze ve velkých data setech a databází hledat vzory v rámci zkoumaných dat, predikovat pravděpodobné výsledky či tvořit užitečné informace.

Grafové databáze jsou speciálním typem NoSQL databází, kde jejich datovým modelem je graf poskytující zajímavý způsob propojení dat pomocí tzv. vztahů. Spolu s použitím metod dataminingu mohou poskytovat výsledky pro další využití v různých oborech i v grafické podobě, která je snadná pro interpretaci.

1.1 Cíl práce

Cílem této bakalářské práce je seznámení s NoSQL databázemi, jakožto jinou možností práce s daty, než jsou relační databáze, a problematikou dataminingu. Se soustředěním na grafické databáze, kde je provedena instalace jedné z takových databází, konkrétně Neo4j databáze, jsou provedeny metody dataminingu v podobě grafových algoritmů za pomocí dotazovacího jazyka Cypher jako ukázky, jak získat určitá data z Neo4j databáze a interpretovat ony výsledky algoritmů. Dalším z cílů je provedení detailní analýzy volně dostupného elektronického data setu obsahující informace ohledně Panama Papers. Nad daty jsou aplikovány jednoduché dotazy pomocí jazyka Cypher a grafové algoritmy. Výsledky analýzy jsou zhodnoceny pro jejich další možné budoucí využití.

2 NoSQL databáze

NoSQL databáze jsou alternativou pro standardní databázové systémy a označují širokou skupinu nerelačních databází. Termín NoSQL je vysvětlen jako „not only SQL“ (v překladu do češtiny „nejen SQL“), což znamená, že NoSQL databáze neodmítají jazyk SQL, ale není používání jako primární.

2.1 Historie a vznik NoSQL databází

Donedávna mezi nejčastěji používaným způsobem pro ukládání a správu dat patřily relační, objektové, objektově relační, XML a další databázové systémy. Nejpopulárnějším systémem je relační databáze, pod kterou si představujeme tabulku s řádky a sloupci. Databázové systémy tohoto typu používají architekturu typu klient/server, z toho vyplývá, že data jsou uložena na jednom serveru a klient k nim přistupuje pomocí klientských programů. V okamžiku, kdy klient potřebuje uložit velké množství dat, server musí zvýšit svou diskovou kapacitu (Holubová, Kosek, Minařík, & Novák, 2015). Pro zpracování Big Data přístupy těchto databázových systému nejsou vyhovující, a je proto nutné přijít s novými přístupy – NoSQL databáze.

První, kdo pravděpodobně přišel s pojmem NoSQL byl Carlo Strozzi, který ve své článku (1998) tak pojmenoval open source relační databázi nepodporující dotazový jazyk SQL.

Později v 21. století v průběhu Internetové éry a vzniku Big Data velké firmy jako Google, Amazon a Facebook apod. se potýkaly s problémy použití relačních databází a jejich omezení kvůli velikosti dat. Společnost Google potřebovala ukládat velké množství obsahů stránek a jejich propojení s odkazy. Řešením bylo vytvoření komprimovaného proprietárního systému BigTable pro ukládání dat. BigTable oproti relačním databázím nebyl řádkově orientovanou databází ale naopak sloupcově orientovanou. Společnost Amazon se potýkala s podobným problémem a přišla s databází Dynamo, která funguje na principu klíč-hodnota (Hřivna, 2016). S těmito východisky začal rozvoj NoSQL databází.

2.2 Porovnání relačních a NoSQL databází

Data spolu s datovou integritou v relační databázi se snadno udržují, také nemají vysoký objem, jelikož nejsou redundantní. Pokud však je potřeba provést změny v celé databázi v podobě přidání určitého atributu, mění se schéma v dané databázi. S možností, jak se přizpůsobit změnám v databázi přicházejí různé typy NoSQL databází.

Relační a každý typ NoSQL databází má své výhody a nevýhody. Představují příležitosti, jak data uložit a následně s nimi pracovat.

Tabulka (tabulka 1) porovnává relačních a NoSQL databáze a jejich předpoklady o datech a jejich zpracování.

Tabulka 1 Porovnání relačních a NoSQL databází (Holubová et al., 2015)

Relační databáze	**NoSQL databáze**
Integrita dat je důležitá.	Integrita dat není příliš zásadní.
Datový formát je soudržný a přesně definovaný.	Datový formát nemusí být soudržný či nemusí být vůbec známý.
Předpokládá se dlouhodobé uložení dat.	Kvůli velkému množství dat se ukládají data pouze v určitém časovém úseku.
K aktualizaci dat dochází často.	Vložená data nejsou později nijak modifikována. Data pouze přibývají a nepotřebná data jsou odstraněny.
Lineární nárůst velikosti dat.	Exponenciální nárůst velikosti dat.
Probíhají pravidelné zálohy dat.	Data nejsou zálohována, ale replikována.
Data většinou uchovává jediný server.	Data jsou většinou umístěna na více serverech.

2.3 Základní principy NoSQL

2.3.1 Databázová transakce (ACID)

Obecně relační databázový systém funguje na principu tzv. transakcí. Transakce jsou tvořeny posloupností operací, jež mají za úkol převádět data z jednoho konzistentního stavu do stavu druhého. Pokud se v průběhu převádění dat vyskytne chyba, data se vrací do svého původního stavu (Hills, 2016).

Transakce mají určité vlastnosti nazývající se ACID – Atomicity, Consistency, Isolation, Durability:

- Atomicity (atomicita) – vyjadřuje nedělitelnost transakce. Transakci je potřeba provést jako nedělitelný celek, v jiném případě není provedena,

- Consistency (konzistentnost) – zaštituje převedení z jednoho konzistentního stavu dat do druhého konzistentního stavu,
- Isolation (izolovanost) – transakce jsou na sobě nezávislé. Operace vně transakcí jsou skryté před ostatními probíhajícími transakcemi,
- Durability (trvanlivost) – zabezpečení výsledků transakcí. Výsledná data jsou uložena do databázového systému (Robinson, Webber, & Eifrem, 2013).

Výše zmíněné vlastnosti transakcí nám zaručují nenarušení konzistenci dat v databázovém systému. Např. při výpadku systému je zaručeno, že nedojde ke ztrátě dat.

2.3.1.1 ACID vs. BASE

V většině případů NoSQL databází není ACID plně využíván, nýbrž jeho opak, což je BASE – Basically Available, Soft state, Eventual consistency:

- Basically Available (převážná dostupnost) – umožňuje přístup k neúplným datům i v případě částečného výpadku systému,
- Soft state (volný stav) – systém je dynamický, dochází tedy k neustálému přepisování dat,
- Eventual consistency (občasná konzistence) – systém může být v konzistentním stavu, ale ovšem tento stav není zaručen neustále (Holubová et al., 2015).

BASE poskytuje občasnou konzistenci, která zpřístupňuje škálovatelnost. Škálovatelnost umožňuje v případě zpracování dat flexibilní reagování na zvyšující se množství dat, které je potřeba ukládat do databází.

Občasná konzistence nabízí řešení pro potřebu rychlého ukládání dat, protože ne vždy je důležité, aby data byla plně konzistentní (např. aktuální či správnost).

2.3.2 Distribuce

NoSQL zpracovává velké množství dat v rámci jednoho a více databázových serverů. V případě více než jednoho databázového serveru existují dvě techniky, které lze i kombinovat:

- rozdělení (sharding) – na různé uzly v clusteru[1] rozmístíme určité části dat (shards),
- replikace – pro zvýšenou dostupnost se data replikují na více uzlů clusteru.

[1] množina síťově propojených počítačů

Je třeba zmínit, že distribuce v grafových databázích nemusí být vždy uskutečněna, a to kvůli své obtížnosti. Pokud je graf úplný[2], není možné jej distribuovat.

2.3.2.1 Rozdělení dat (sharding)

Data můžeme rozdělit na určité celky a ukládat na různé uzly clusteru. Uživatelé při vyhledávání nepřistupují k jedinému serveru, ale více, podle toho, o jaké data mají zájem. Strategie rozdělení je klíčová a určuje, jak bude systém efektivní.

Rozdělování má typicky cíl, jenž je v zájmu dosažení. K dispozici je dosažení rovnoměrného umístění dat na uzlech či minimalizovat počet daných uzlů anebo optimalizovat rozmístění dat v rámci geografie. Cíle se navzájem vylučují a je proto důležité dosáhnout jistého kompromisu (Panyko, 2013). Například v rámci výpadku sítě uživatelé ztrácí přístup k určitým datům, rozdělení dat proto bývá často spojeno s replikací.

2.3.2.2 Master-slave replikace

Správce systému má možnost určit, který z uzlů bude primární (master) a jaké budou sekundární (slaves) za podmínky, že data v tomto systému budou spíše používána pro čtení než modifikaci.

Veškeré požadavky na zápis či popřípadě změnu probíhají na master uzlu, následně primární uzel informuje své sekundární uzly o provedených změnách (replikace dat) (Tiwari, 2011). V případě selhání master uzlu, jakýkoliv slave uzel může převzít status primárního uzlu, jelikož data na veškerých uzlech jsou stejná, ovšem zpracovává požadavky pouze na čtení, dokud správce systému plnohodnotně neručí nový master uzel.

2.3.2.3 Peer-to-peer replikace

Peer-to-peer replikace dat reaguje na omezení, které se sebou nese master-slave replikace. Pokud totiž dojde v peer-to-peer k výpadku určitého uzlu, ostatní uzly neztrácejí možnost čtení a zapisování, kvůli rovnocennosti všech uzlů. Ovšem peer-to-peer replikace má také nevýhodu v podobě hrozby, která vzniká při dvou současných operací přepisování stejných dat. Hrozba může trvala poškodit data v systému.

[2] mezi většinou dvojic uzlů vede hrana

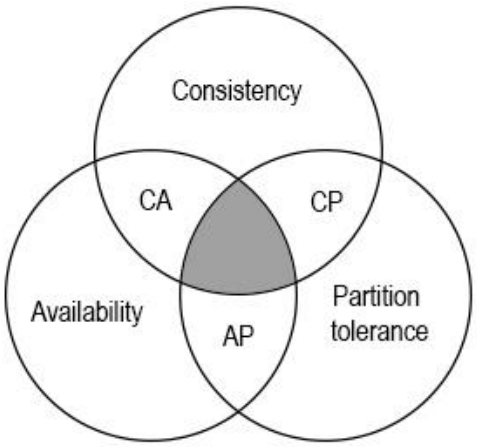
2.3.2.4 Replikace a sharding

Jak bylo výše zmíněno, pro efektivní distribuci dat se používá kombinace sharding a replikace (Lažanský, 2014). Aby tohoto bylo dosaženo, nejprve se dělí data podle určité strategie a poté se volí jeden z možných typů replikace (master-slave, peer-to-peer).

2.3.3 CAP teorém

Na distribuovaný databázový systém lze velmi obtížně aplikovat ACID vlastnosti. Používá se jiný přístup zvaný CAP teorém, jindy také Brewerův teorém podle svého autora Erica Brewera. Zmíněný teorém zaštituje chod distribuovaných systémů těmito vlastnostmi:

- Consistency (konzistence) – všechny uzly distribuované databáze v určitém čase mají k dispozici aktuální verzi dat,
- Availability (dostupnost) – veškeré požadavky na čtení a zápis systém vždy obslouží,
- Partition tolerance (odolnost vůči rozpadu sítě) – v případě částečnému výpadku sítě, kdy se systém rozdělí na několik izolovaných částí, je systém stále schopný zpracovávat požadavky na něj kladené (Šeleng, Laclavík, Dlugolinský, & Hluchý, 2011).



Obrázek 1 CAP teorém (Holubová et al., 2015)

Splnění všech tří vlastností současně však nelze dosáhnout. Na obrázku (obrázek 1) je CAP teorém znázorněn a ukazuje, že distribuovaný systém může splňovat pouze dvě vlastnosti. Nejpříhodnějšími variantami jsou CP a AP. CA kombinace není doporučována, protože distribuovaný systém bez odolnosti vůči rozpadu sítě se nesnadno realizuje a je prakticky nevyužitelný.

2.4 Typologie NoSQL databází

V současné době je NoSQL databáze nesnadné zařadit do jedné specifické kategorie, ovšem podle základního rozdělení s podobným chováním a datovým modelem rozlišujeme čtyři základní typy: databáze typu klíč-hodnota, dokumentové, sloupcové a grafové databáze (Vardanyan, 2017).

2.4.1 Databáze typu klíč-hodnota

Databázový systém typu klíč-hodnota je nejjednodušší typ NoSQL databází. Data jsou ukládaná do databáze na základě jejich jedinečných klíčů, což umožňuje snadné a efektivní vyhledávání. Ovšem vyhledávání bez daného klíče není možné. Používají se například pro ukládání mezi-paměti, protokolů apod. (Škrášek, 2015).

2.4.2 Dokumentové databáze

Dokumentové databáze se velmi podobají databázovým systémům typu klíč-hodnota. Jejich primárním rozdílem je způsob ukládání dat, jelikož v dokumentové databázi mají data určitou strukturu a kódování. Typickým příkladem jsou formáty JSON, XML a binární formáty jako např. PDF (Bartha, 2015). Každý dokument má metadata, podle kterých lze dokument zařadit do podobné skupiny dokumentů, umožňující snadné vyhledávání podle jejich obsahu.

2.4.3 Sloupcové databáze

Pro zpracování velkého množství dat horizontálně distribuované mezi několika servery se používají sloupcové databáze. Používají některé z pojmů relačních databází – sloupec a řádek. Jejich princip však není úplně stejný. Sloupcové databáze umožňují do každého řádku vkládat sloupce, aniž by to dále ovlivňovalo řádky ostatní. Při vytváření datového modelu databáze je důležité, aby byl promyšlený, jelikož na něm závisí budoucí výkon dané databáze (Řáda, 2017).

2.4.4 Grafové databáze

Grafové databáze se výrazně liší od zmíněných databází výše. Databáze se používají pro data, která mají mezi sebou vzájemné vazby. Jak už název napovídá, hlavním aspektem, tedy daty, jsou grafy. Lze je vhodně modelovat a následně se na ně dotazovat – jak na grafy, tak i na vazby mezi nimi.

3 Grafové databáze

Jedním z typů NoSQL databází jsou grafové databáze. Grafové databáze byly navrženy s ohledem na to, že vývojáři často ve svých aplikacích vytvářejí grafické struktury, ale i přesto si data nepřetržitě ukládali buď v relačních tabulkách nebo v různých typech NoSQL databází. Grafové databáze nacházejí své uplatnění v různých oblastech, jako např. sociální sítě a systémy pro doporučení, kde použití relačních databází není možné.

Datový model grafu (uzly, hrany, vlastnosti) je jádrem databází grafů a může představovat spoustu komplexních softwarových požadavků a efektivita a výkon traverzového[3] dotazování grafů jsou hlavními přednostmi databází grafů.

3.1 Typy grafů

Podstatou grafových databází jsou grafy, které z hlediska teorie grafů můžeme dělit na orientované a neorientované. Orientovanou hranou se rozumí směr hrany, z jakého uzlu a do jakého uzlu hrana směruje. Určuje se tím tak počáteční a koncový uzel orientované hrany. Naopak neorientovaná hrana směřuje oběma směry mezi uzly. Může také vzniknout graf tzv. smíšený, kde některé uzly mezi sebou mají orientované hrany a jiné neorientované.

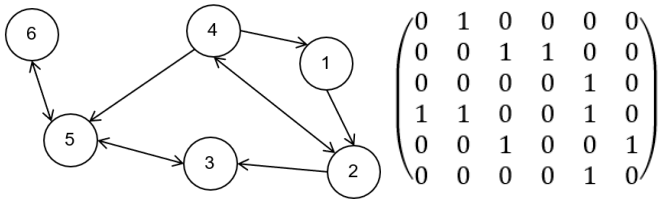
Rozlišujeme také jednovztahové a vícevztahové grafy. Pokud se jedná o jednovztahový graf, všechny hrany mají homogenní nespecifikovaný typ, ve vícevztahovém grafu hrany mají různé typy vyjadřující heterogenní vztahy. Hrany vyjadřující různé vztahy můžou tvořit multigrafy. Grafy, kde mezi dvěma uzly může vést více hran. Hypergrafy umožňují vyjadřovat vztahy mezi více než dvěma uzly. Uzly a hrany navíc můžou mít i další atributy, jedná se potom o atributové grafy.

[3] iterativní procházení částí grafů

3.2 Reprezentace grafů

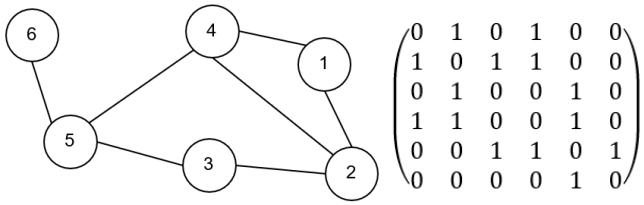
3.2.1 Matice sousednosti

Orientovaný a neorientovaný graf lze v poli lze uložit pomocí matice sousednosti. Matice obsahuje booleovské hodnoty, kde počet sloupců a řádků v matice se rovná počtu uzlů grafu.



Obrázek 2 Reprezentace orientovaného grafu pomocí matice sousednosti (Zdroj vlastní.)

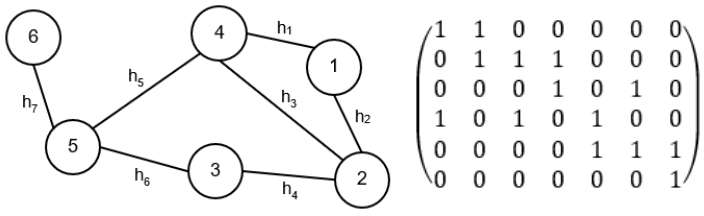
V případě orientovaného grafu je matice vždy čtvercová (obrázek 2). Pro graf neorientovaný není nutné ukládat matici celou, ale jen trojúhelníkovou matici, jelikož části pod a nad diagonálou jsou identické (obrázek 3).



Obrázek 3 Reprezentace neorientovaného grafu pomocí matice sousednosti (Zdroj vlastní.)

3.2.2 Incidenční matice

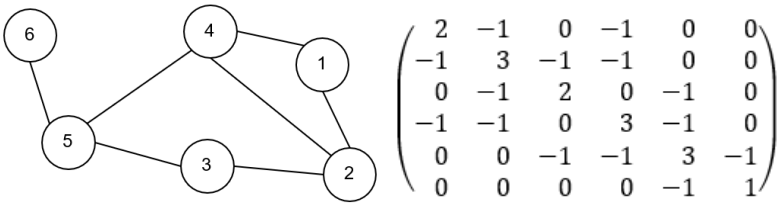
Obrázek (obrázek 4) znázorňuje matici incidence a neorientovaný graf. Matice incidence se skládá z *n* řádků a *m* sloupců, kde sloupec představuje hranu a hodnoty *1* označují uzly tvořící *h* hranu, a řádek představuje uzel, kde hodnoty *1* označují všechny hrany patřící k danému uzlu. Ve většina případů platí, že *m* je větší než *n* (“Reprezentace grafů,” n.d.).



Obrázek 4 Reprezentace neorientovaného grafu pomocí matice incidence (Zdroj vlastní.)

3.2.3 Laplaceova matice

Laplaceova matice je čtvercová matice, kde diagonála vyjadřuje stupeň uzlu, tj. počet hran jejíchž je prvkem.

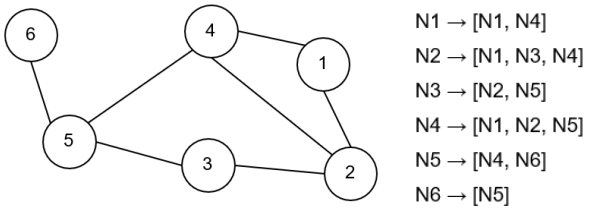


Obrázek 5 Reprezentace neorientovaného grafu Laplaceovou maticí (Zdroj vlastní.)

Hodnota -1 v matici pro daný uzel zaznamenává jeho sousední uzly. Na obrázku (obrázek 5) je zobrazen neorientovaný graf spolu s Laplaceovou maticí.

3.2.4 Seznam sousedů

Graf lze reprezentovat i pomocí tzv. seznamu sousedů (obrázek 6), který má každý vrchol. Zaznamenání seznamu odpovídá vektoru *n* ukazatelů.



Obrázek 6 Reprezentace neorientovaného grafu pomocí seznamu sousedů (Zdroj vlastní.)

3.3 Významné grafové databáze

3.3.1 Neo4j

Neo4j je systém pro správu grafových databází vyvinutí společnosti Neo4j. Jedná se o transakční databázi kompatibilní s ACID. Neo4j je implementován v Javě. Dotazovacími jazyky jsou Cypher, Gremlin a Java API.

3.3.2 ArangoDB

ArangoDB je vyvinut společností ArangoDB a jedná se multi-modelový databázový systém, podporující dokumentový, grafový a klíč-hodnota model. Dotazovacím jazykem je pouze AQL, který je velmi podobný jazyku SQL (Yuhanna, 2017).

3.3.3 OrientDB

OrientDB je open source systém pro správu databází NoSQL napsaný v jazyce Java. Jedná se o databáze podporující multi-model, tj. podporování grafových,

dokumentových a klíč-hodnota modelů. OrientDB je založena na principu grafové struktury, tedy na přímém spojení mezi jednotlivými záznamy. Mimo jiné dotazování je pomocí jazyka SQL, ovšem bez možností `JOIN`.

3.3.4 Amazon Neptune

Amazon Neptune je grafový databázový produkt od společnosti Amazon používaný jako webová služba a je součástí služby Amazon Web Services. Dotazovacím jazykem je Gremlin a SPARQL (Yuhanna, 2017). Amazon Neptune se většinou používá pro doporučování kontentu, detekci podvodů a zabezpečení sítě.

4 Neo4j databáze

Grafový databázový systém Neo4j vytvořen společností Neo4j Inc., patří do skupiny NoSQL databází. Vzhledem k tomu, že je implementován v jazyce Java, je přenositelný mezi operačními systémy.

Neo4j databáze v rámci bakalářské práce byla vybrána z důvodů, že se jedná o open-source databázi, má velmi přehlednou dokumentaci a vizualizační prostředí je uživatelsky příjemné.

4.1 Základní charakteristika

Neo4j oproti ostatním NoSQL databázím splňuje ACID vlastnosti, které se týkají transakcí. Všechny editace grafu probíhají v transakci. Dále také nabízí vysokou dostupnost (Neo4j High Availability), která je založena na master-slave architektuře. Neo4j může tedy pracovat na jednom serveru či v clusteru uzlů (Vukotic & Watt, 2015).

Neo4j využívá indexaci, kde indexy jako datová struktura slouží pro efektivnější vyhledávání uzlů a hran grafů. Index je jedinečné jméno, které zadává uživatel a umožňuje mu tak asociovat libovolné množství dvojic (klíč-hodnota) s libovolným množstvím indexovaných uzlů nebo hran.

4.2 Použití Neo4j databáze

Grafové databáze, stejně jako ostatní typy NoSQL databází, jsou aplikovány na specifické typy problematik. I přesto, že se jedná o velmi mladou technologii, svoje uplatnění nalezla v různých oblastech (“Graph Database Use Cases and Solutions,” n.d.):

- analyzování datových vztahů v reálném čase pro odhalování podvodných kruhů a dalších sofistikovaných podvodů,
- řízení sítí a IT infrastruktury,
- personalizace produktů, doporučení obsahů a služeb v reálném čase,
- využívání deklarovaných sociálních kontaktů či vyvozování vztahů na základě aktivit v rámci sociálních sítí,
- sledování uživatelů, majetků, vztahů a oprávnění při správě totožnosti a přístupu.

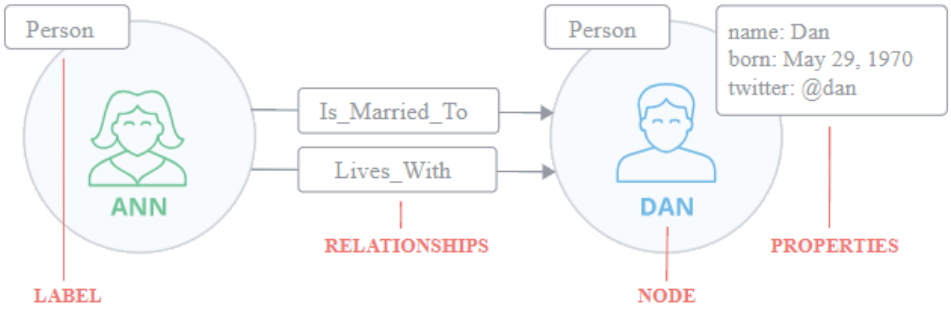
Americká maloobchodní společnost provozující řetězce diskontních obchodů Walmart začala používat grafovou databází Neo4j, aby porozuměla chování spotřebitelů nakupující v e-shopu a byla schopná v reálném čase nabízet zákazníkovi podobné produkty či

produkty, které by ho mohli zajímat na základě jeho aktivit na e-shopu (“Walmart - Neo4j Graph Database Platform,” n.d.).

Další americká společnost využívající Neo4j je eBay, která využívá grafovou databází pro sledování svých doručovatelů a zásilek v reálném čase (“Walmart and eBay adopt graph database | News | Retail Technology,” n.d.).

4.3 Datový model

Grafová databáze Neo4j je založena na grafové struktuře. Datový model (obrázek 7) se tedy skládá z uzlů (nodes), hran (relationships) a vlastností (properties), pak existují tzv. štítky (labels).



Obrázek 7 Datový model grafové databáze Neo4j (“Neo4j Graph Platform – The Leader in Graph Databases,” n.d.)

Uzly (nodes) jsou hlavními datovými prvky, které slouží k vyjadřování entit. Jednomu a více uzlům lze přiřadit štítek (label) pro rozlišení a přiřazení do skupin. Štítky jsou velmi důležité a efektivní pro práci pouze z částí grafu namísto celého (Vukotic & Watt, 2015).

Spojení mezi uzly znázorňují orientované či neorientované hrany (relationships). Hrana má vždy začátek a konec a může vést od jednoho uzlu do druhého či vytváří smyčku, kde hrana odkazuje na stejný uzel.

Vlastnosti (properties) lze přiřadit k jednotlivým uzlům a hranám. Bývají definovány jako klíč-hodnota a uchovávají metadata či přídavné informace.

4.4 Instalace Neo4j databáze

4.4.1 Licence

Neo4j je open source projekt, jehož zdroje a autorská práva vlastní a udržuje společnost Neo4j Inc. Společnost Neo4j disponuje s dvěma typy licencí. Verze Community pro

nekomerční užití je pod licencí GNU GPLv3 a verze Enterprise již pro komerční účely je pod licencí GNU AGPLv3, jak popisuje tabulka (tabulka 2).

Tabulka 2 Přehled licencí poskytující Neo4j ("Neo4j Licensing Overview," n.d.)

Verze	Stručný popis	Licence
Community	základní databáze, podporující ACID	GNU GPLv3
Enterprise	online zálohování, clustering, vysoká dostupnost	GNU AGPLv3

4.4.2 Instalace Neo4j Community Serveru

Pro účely bakalářské práce je provedena instalace produktu Neo4j Community Serveru. Společnost Neo4j, Inc poskytuje instalační soubor pro instalaci grafové databáze Neo4j Community Server na svých webových stránkách ("Neo4j Graph Platform – The Leader in Graph Databases," n.d.). V době instalace byla dostupná verze Neo4j Community Serveru 3.5.2. Důležité systémové požadavky, jež jsou potřeba pro provoz Neo4j Community Serveru jsou zmíněné spolu s používanými v tabulce (tabulka 3).

Tabulka 3 Systémové požadavky pro instalaci Neo4j Community Serveru ("2.1. System requirements - Chapter 2. Installation," n.d.)

	Požadavky		
	Minimální	**Doporučené**	**Používané**
Procesor	Intel Core i3	Intel Core i7; IBM POWERS8	Intel Core i5
Paměť	2 GB	16-32 GB	8 GB
Disk	10 GB SATA	SSD w/ SATA Express nebo NVMe	SSHD SATA
Souborový systém	EXT4 (či podobný)	EXT4, ZFS	NTFS
Operační systém	Windows Server 2012, 2016; Ubuntu 18.04, 16.04, 14.04; Debian 8, 9; CentOS 6, 7; Fedora; Red Hat; Amazon Linux		Windows 10

Než však bude Neo4j Community Server nainstalován, je potřeba nejdříve nainstalovat OpenJDK 8 (Open Java Development Kit), což je open source implementace Java Platform Standard Edition (Java SE).

Pro využívání vizualizačního nástroje Neo4j Browser je nutné nainstalovat službu přes příkaz v příkazovém řádku v adresáři Neo4j Community Serveru:

```
bin\neo4j install-service
```

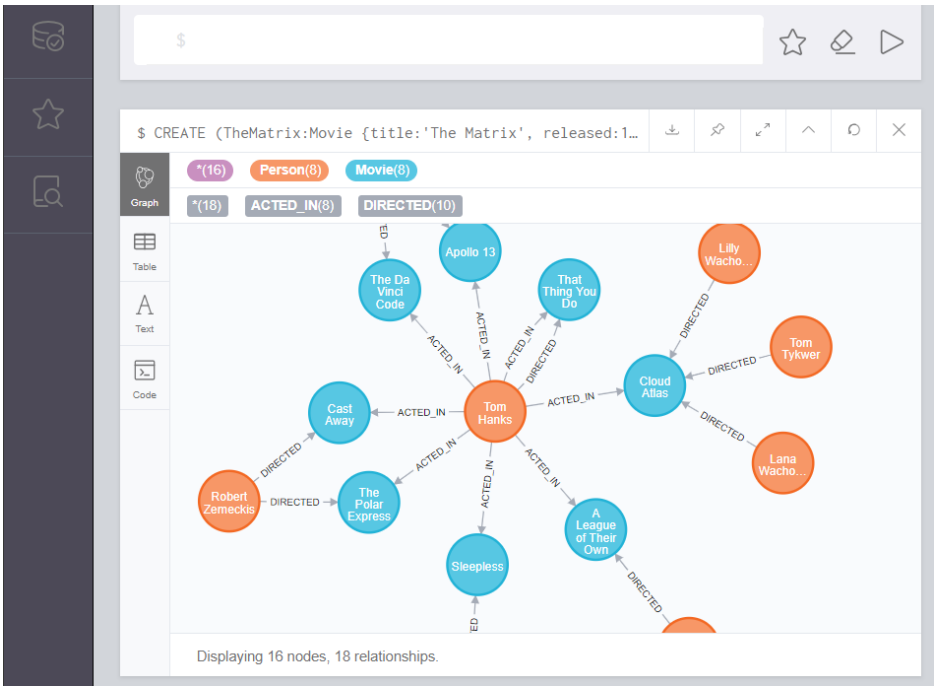
Pro spuštění Neo4j Browseru zadáváme do webového prohlížeče defaultní adresu:

```
http://localhost:7474/
```

Po provedení výše zmíněných krocích je nyní možné pracovat s Neo4j databází.

4.5 Vizualizační nástroj Neo4j Browser

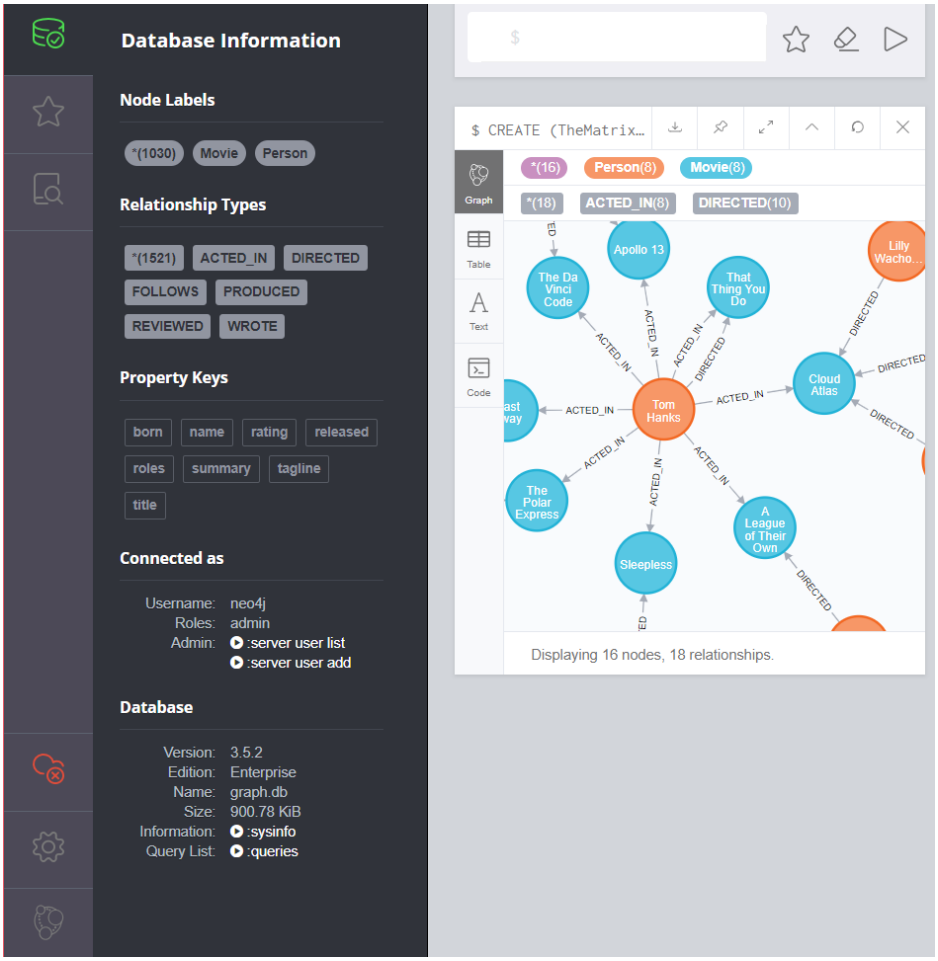
Neo4j Browser lze použít pro přidávání dat, spouštění dotazů, vytváření vztahů apod. Umožňuje také snadnou vizualizaci dat v databázi.



Obrázek 8 Vizualizační nástroj Neo4j Browser (Zdroj vlastní.)

Na obrázku (obrázek 8) lze vidět prostředí Neo4j Browseru. Dotazy a příkazy pro vytváření a načítání dat se zadávají do editoru, který je na obrázku vyobrazen nahoře uprostřed, napravo se pomocí šipky spouští operace. Pod editorem je stream, kde se zobrazují výsledky dotazů. Na postranním panelu streamu má uživatel možnost, jak si data zobrazit.

Defaultně se zobrazuje graf, dále pak kliknutím např. na ikonu Table se data zobrazí v podobě tabulky, která zobrazuje uzly a hrany.



Obrázek 9 Vizualizační nástroj Neo4j Browser – postranní panel (Zdroj vlastní.)

Obrázek (obrázek 9) zobrazuje rozkliknutí postranního panelu v Neo4j Browseru. Postranní panel nabízí různé možnosti jako je prohlížení detailů databáze, prohlížení či změna nastavení Neo4j Browseru, prohlížení dokumentace a další. Kliknutím na jednu z možností se zobrazí širší posuvný postranní panel s podrobnostmi o dané možností.

4.6 Neo4j Sandbox

Neo4j Sandbox poskytuje pohodlný způsob interakce s instancí Neo4j v kombinaci online návodů. Sandbox nabízí široký výběr různých populárních případů použití, které obsahují samotné soubory dat a mimo jiné i návody pro dotazování a vizualizaci dat. Mezi případy užití patří např. Panama Papers, Paradise Papers, Russian Twitter Trolls, Legis-Graph apod. Sandbox lze zpřístupnit pomocí Neo4j Browseru.

4.7 Přístup k Neo4j databázi

Grafová databáze Neo4j umožňuje několik možností pro procházení grafem. Jednou z možností je přes vestavěné rozhraní Java API (Neo4j je implementován v Javě), které umožňuje přístup z programového kódu. Dále existují dotazující jazyky přímo pro práci s grafy – Gremlin a Cypher.

4.7.1 Gremlin

Gremlin je doménově specifikovaný jazyk, který slouží k načtení dat z grafu a jejich úpravě. Gremlin se jako jazyk orientuje na cestu, který stručně vyjadřuje komplexní přechody grafů a mutační operace. Dotazovací jazyk je imperativní a umožňuje vykonávat traverzování grafů (Vukotic & Watt, 2015).

4.7.2 Cypher

Cypher byl navržen přímo pro Neo4j databázi. Oproti dotazovacímu jazyku Gremlin se jedná o deklarativní jazyk. Místo způsobu, jak projít graf, umožňuje uživateli, co může průchodem grafu získat. Cypher má velké podobnosti s dotazovacím jazykem SQL, a to tvoří Cypher uživatelsky jednoduchý a přehledný.

Mezi základní příkazy patří (Vukotic & Watt, 2015):

```
START // určení počátečních uzlů grafu
MATCH // vzor navázaný na počáteční uzly, kterému musí požadovaný graf odpovídat
WHERE // filtrovací kritéria
RETURN // návratové hodnoty
CREATE // vytváření uzlů a hran
DELETE // mazání uzlů, hran a vlastností
SET // nastavení/změna hodnot uzlů a hran
FOREACH // provedení změn nad všemi prvky daného seznamu
ORDER BY // seřazení výsledku podle zvoleného kritéria
MERGE // zajišťuje, že v grafu existuje vzorec, pokud ne, je třeba jej vytvořit
UNION // sloučí výsledky dvou nebo více dotazů
```

4.7.2.1 Použití Cypheru

Použití dotazovacího jazyka Cypher v Neo4j Browseru lze představit na jednoduchém příkladu. Na příkladu bude znázorněno, jak vytvořit jednotlivé uzly a hrany. Pomocí příkazů dále bude odstraněn uzel a jeho příslušné hrany.

Pokud chceme vytvořit jednoduchý uzel použijeme následující příkaz, kde g představuje proměnou týkající se pouze v rámci daného příkazu:

```
CREATE (g:Game { Name: "Dishonored" });
```

Vzhledem k tomu, že není vrácena žádné proměnná, Neo4j Browser vrací zprávu o úspěchu, která říká, že byl přidán 1 uzel s 1 vlastností a 1 štítkem.

Chceme dále vytvořit další uzel, který bude mít 2 vlastnosti. Následující příkaz vrací proměnou p, tudíž se vykreslí graf (obrázek 10):

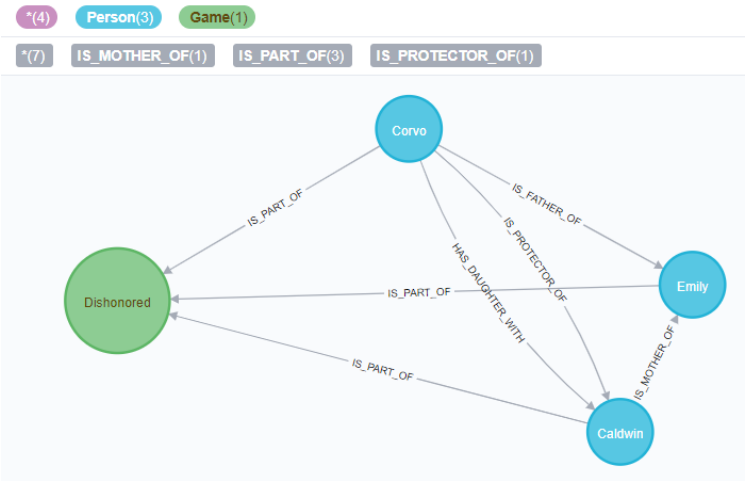
```
CREATE (p:Person { Name: "Corvo", Title: "Royal Protector"})
RETURN p;
```

Obrázek 10 Vytvoření uzlu grafu pomocí dotazovacího jazyka Cypher (Zdroj vlastní.)

Pomocí příkazu `CREATE` vytvoříme další 2 uzly (Person) – *„Emily“* a *„Caldwin“*. Mezi uzly jsou jisté vztahy (hrany). Hrany grafu vytvoříme následovně:

```
MATCH (g:Game),(p1:Person), (p2:Person), (p3:Person)
WHERE g.Name = "Dishonored" AND p1.Name = "Corvo"
AND p2.Name = "Emily" AND p3.Name = "Caldwin"
CREATE (p1)-[:IS_PART_OF]->(g), (p2)-[:IS_PART_OF]->(g), (p3)-
[:IS_PART_OF]->(g), (p1)-[:IS_FATHER_OF]->(p2),
(p3)-[:IS_MOTHER_OF]->(p2), (p1)-[:HAS_DAUGHTER_WITH]->(p3),
(p1)-[:IS_PROTECTOR_OF]->(p3)
RETURN p1,p2,p3,g;
```

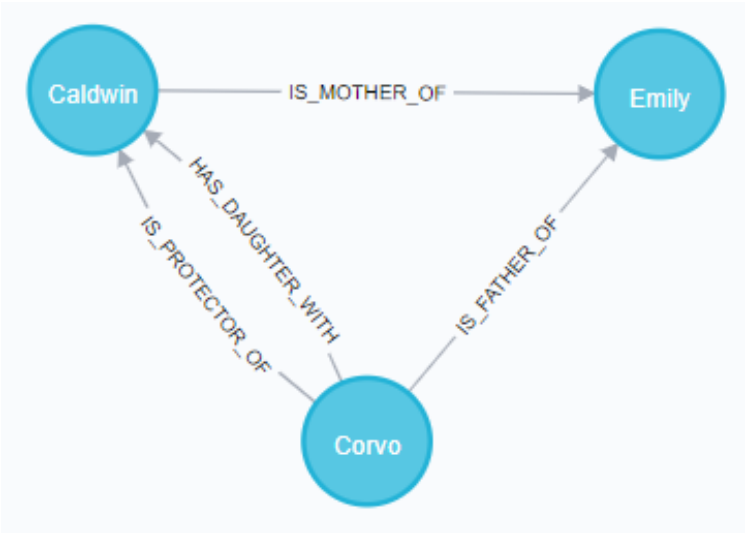
Neo4j Browser zobrazí (obrázek 11) celý graf (všechny uzly a jejich hrany).



Obrázek 11 Vytvoření hran grafu pomocí dotazovacího jazyka Cypher (Zdroj vlastní.)

Pokud chceme odstranit uzel/uzly je potřeba nejprve odstranit veškeré hrany, které se k danému uzly vážou. Pokud bychom chtěli odstranit uzel Game nazývaný *„Dishonored“*, odstraní se nejprve hrana s názvem *„IS_PART_OF“*, až poté lze uzel odstranit. To lze provést pomocí příkazu následovně:

```
MATCH ()-[r:IS_PART_OF]-() DELETE r;
MATCH (g:Game) DELETE g;
MATCH (n) RETURN n;
```



Obrázek 12 Odstranění uzlu a hran pomocí dotazovacího jazyka Cypher (Zdroj vlastní.)

Výsledný graf (obrázek 12) se po provedení příkazu skládá už pouze jen z uzlů *„Person“* a jejich příslušných hran.

5 Datamining v grafové databázi Neo4j

Datamining (v češtině dolování dat) je praxe automatického prohledávání velkých uložišť dat pro objevení vzorů a trendů, které jsou nad rámec jednoduché analýzy ("Data Mining Concepts," n.d.). Dolování dat využívá sofistikované matematické algoritmy pro segmentaci dat a vyhodnocení pravděpodobnosti budoucích událostí. Datamining je částí dobývání znalostí z databází (KDD).

Klíčové vlastnosti dataminingu:

- automatická detekce vzorů,
- predikce pravděpodobných výsledků,
- tvorba užitečných informací,
- zaměření na velké data sety a databáze.

Data mohou být dolována ze souborů, tabulek, různých typů databází apod., formát v jakém jsou data není tak důležitý jako jeho použitelnost na daný problém, který má být pomocí dataminingu vyřešen. V rámci dataminingu se využívá termín *Data Warehouse* (v češtině datový sklad) vyjadřující uložiště dat pro datamining. Důležité je, aby datový sklad obsahoval potřebná data.

Pomocí dataminingu se odkrývají vzory a vztahy v rámci zkoumaných dat – odhalují se jím skryté informace v datech.

Datamining se nejčastěji využívá v rámci detekci podvodů, finančního bankovnictví, segmentaci zákazníků, kriminálních vyšetřování, různých výzkumných analýz apod. (Rajkumar, 2014).

5.1 Grafové algoritmy

Grafové algoritmy se používají k výpočtu metrik pro grafy, uzly nebo hrany. Mohou poskytnout pohled na relevantní entity v grafu (centrality pořadí) nebo inherentní struktury, jakou jsou komunity (komunitní detekce, rozdělení grafů a clustering[4]).

Mnoho grafových algoritmů jsou iterativního přístupu, které procházejí grafem pro zjištění náhodné procházky, prohledávání do šířky, prohledávání do hloubky a nebo shody vzorů ("Chapter 1. Introduction - The Neo4j Graph Algorithms User Guide v3.5," n.d.).

[4] shlukování, propojování

Vzhledem k exponenciálnímu růstu možných cest se vzrůstající vzdáleností má mnoho přístupů také vysokou algoritmickou složitost.

V rámci grafové databáze Neo4j a zásuvného modelu neboli knihovnou *Grafové algoritmy* rozeznáváme následující grafové algoritmy:

- centralities (centrálnosti) – algoritmy pro určení významu jednotlivých uzlů v síti,
- community detection (detekce společenství) – algoritmy pro vyhodnocení, jak je skupina seskupena nebo rozdělena. stejně jako její tendence posilovat nebo rozdělit se,
- path finding (hledání cesty) – algoritmy pro hledání nejkratší cesty v grafu nebo zhodnotit dostupnost a kvalitu cest,
- similarity (podobnost) – algoritmy pro vypočítávání podobnosti uzlů,
- link prediction (predikce propojení) – algoritmy pro určení blízkosti dvojici uzlů,
- preprocessing (předběžné zpracování) – algoritmy pro transformaci dat pro další použití datového toku.

5.2 Instalace grafových algoritmů v rámci Neo4j

Pro práci s grafovými algoritmy v rámci Neo4j je nutné instalovat zásuvný modul *Grafové algoritmy* kompatibilní s verzí Neo4j. Zásuvný modul je soubor volně ke stažení s příponou *.jar* (*Efficient Graph Algorithms for Neo4j. Contribute to neo4j-contrib/neo4j-graph-algorithms development by creating an account on GitHub*, 2017/2019). Po stažení se soubor vkládá do následujícího adresáře:

```
$NEO4J_HOME/plugins
```

Vzhledem k tomu, že algoritmy používají rozhraní API nižší úrovně pro čtení a zápis, z bezpečnostních důvodů je nutné zásuvný model povolit v konfiguraci – přidat do konfiguračního souboru `$NEO4J_HOME/conf/neo4j.conf` následující:

```
dbms.security.procedures.unrestricted=algo.*
```

Po úpravě konfiguračního souboru je třeba restartovat Neo4j. Pro zobrazení seznamu všech algoritmů použijeme následujíc příkaz, který vrací 75 záznamů:

```
CALL algo.list()
```

5.3 Užití grafových algoritmů v Neo4j

Grafová databáze Neo4j v rámci zásuvného modulu nabízí až 75 možných algoritmů, které lze aplikovat na různé typy dat pomocí dotazovacího jazyka Cypher. Následující výčet algoritmů popisuje ty nejběžnější či nejpoužívanější v rámci práce s daty.

5.3.1 Centrálnosti

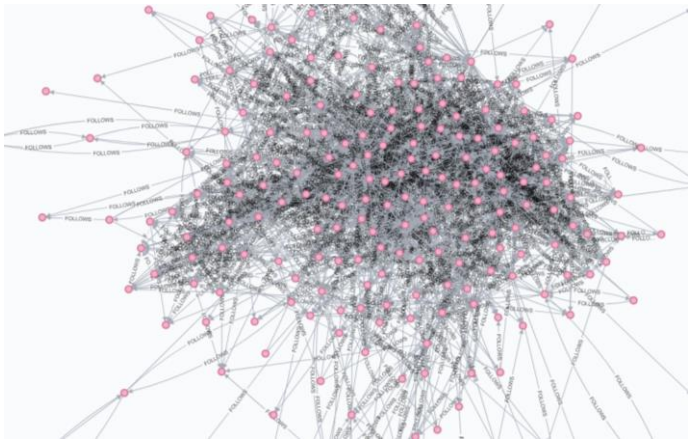
Algoritmy centrálnosti nachází své nejlepší uplatnění při analýze sociálních sítí, vypočítávají důležitost jakéhokoliv daného uzlu v síti. Algoritmy tudíž umožňují pronikat ze změti dat do části sítě, která vyžaduje další pozornost (“Social network analysis,” 2014).

Nejzákladnějším algoritmem centrálnosti je *Degree Centrality* (v češtině Centralita měřená stupněm uzlu). Algoritmus měří počet příchozích a odchozích hran z uzlu, tímto lze najít v grafu nejvíce oblíbené či populární uzly či jinými slovy měří určitou aktivitu uzlů v síti.

Centralitu měřenou stupněm uzlu lze předvést na datech získaných z Internetu týkající se problematiky sociální sítě Twitter (“Network datasets: Social circles,” n.d.). Pro ukázku byl vybrán soubor s uzly znázorňující vztahy mezi uživateli – tj. kdo koho sleduje. Dotaz pro nahrání dat ze souboru je následující:

```
LOAD CSV FROM "file:///12831.edges" AS row FIELDTERMINATOR " "
MERGE (u1:User {id:row[0]})
MERGE (u2:User {id:row[1]})
MERGE (u1)-[:FOLLOWS]->(u2)
```

Graf (obrázek 13) na základě importovaných dat má 236 uzlů a 2 478 hran.



Obrázek 13 Ukázka zobrazení grafu sociální sítě Twitter ze získaných dat z Internetu (Zdroj vlastní.)

Pro vyjádření centrality měřenou stupněm uzlu lze použít následující dotaz:

```
MATCH (u:User)
RETURN u.id AS ID, size(()-[:FOLLOWS]->(u)) AS Followers
ORDER BY Followers DESC
LIMIT 5
```

Za pomocí funkce `size()` byly získány informace zobrazené tabulkou (obrázek 14) o tom, kterých pět osob je nejvíce sledovaných na sociální síti v rámci importovaného souboru.

ID	Followers
"180505807"	52
"1260231"	46
"14231571"	46
"380"	44
"11178592"	42

Obrázek 14 Tabulka zobrazující nejvíce sledovaných uzlů z dat sociální sít Twitter (Zdroj vlastní.)

Dalším zajímavým algoritmem centrálnosti je *Closeness Centrality* (v češtině Blízkost polohy ke středu). Algoritmus detekuje uzly schopné efektivně šířit informace prostřednictvím grafu a měří jejich průměrnou vzdálenost ke všem ostatním uzlům. Uzly s vysokým hodnocením blízkosti mají nejkratší vzdálenost ke všem ostatním uzlům (“4.4. The Closeness Centrality algorithm - Chapter 4. Centrality algorithms,” n.d.).

Blízkost polohy ve středu předvedena na stejných datech (sociální sít Twitter) lze provést pomocí funkce `algo.closeness.stream()`, a to následovně:

```
CALL algo.closeness.stream("User", "FOLLOWS") YIELD nodeId, centrality
RETURN algo.getNodeById(nodeId).id AS ID, centrality AS Centra-
lity_weight
ORDER BY centrality DESC
LIMIT 5
```

Z tabulky (obrázek 15) lze zjistit, že uzly s ID „7899982“ a „13927832“ mají mezi skóre 1, což znamená, že mají přímé spojení s ostatními uzly. Ovšem to neznamená, že mají přímo spojené se všemi uzly v grafu. Zmíněné uzly jsou propojené pouze mezi sebou, tudíž uzel, který je teprve zajímavý v rámci celkových dat je uzel s ID „180505807“, což je zřejmé už jenom z tabulky (obrázek 14), jelikož má přímé spojené s 52 uzly.

ID	Centrality_weight
"7899982"	1.0
"13927832"	1.0
"180505807"	0.5344036697247706
"1186"	0.528344671201814
"652193"	0.513215859030837

Obrázek 15 Tabulka zobrazující blízkost polohy ve středu na datech ze sociální sítě Twitter (Zdroj vlastní.)

5.3.2 Hledání cesty

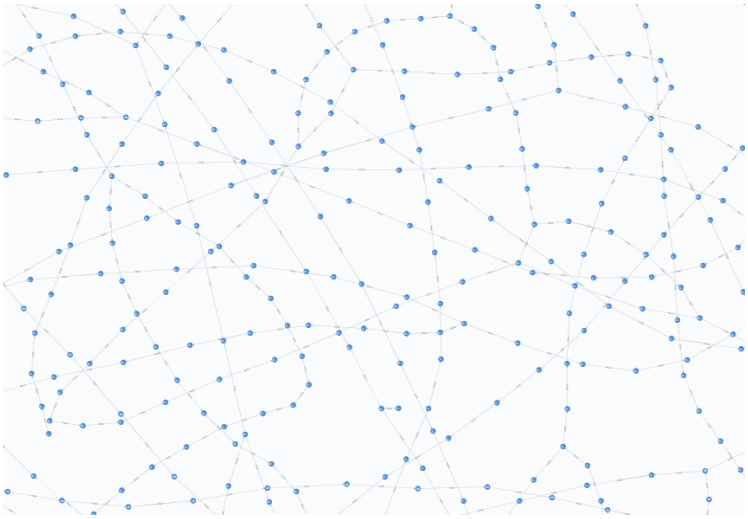
Algoritmy pro hledání cesty zkoumají cesty mezi uzly a vyhodnocují např., která z cest je nejkratší. Některé variace algoritmů hledají nejkratší cestu přičemž je znám startovní uzel a cílový uzel. Své užití nacházejí především při identifikaci optimálních tras grafem jako je plánování logistiky, volání s minimálními náklady, směrování IP adres apod. (Hodler & Needham, 2019).

Při hledání nejkratší cesty mezi uzly se používá např. *Dijkstra Shortest Path algorithm* (v češtině Dijkstrův algoritmus). Dochází k výpočtu nejkratší (vážené) cesty mezi párem uzlů.

Dijkstrův algoritmus na následujícím příkladu je proveden na datech získaných z Internetu týkající se Kalifornské sítě silnic (“Real Datasets for Spatial Databases: Road Networks and Category Points,” n.d.). Soubor zachycuje jednotlivé křižovatky a vzdálenosti mezi nimi pomocí Euklidovské metriky. Dotaz pro import dat ze souboru je následující:

```
LOAD CSV WITH HEADERS FROM "file:///edges.csv" AS line
MERGE (c1:Crossroad {id: line.start_node})
MERGE (c2:Crossroad {id: line.end_node})
MERGE (c1)-[:ROAD {distance: toFloat(line.distance)}]->(c2)
```

Graf (obrázek 16) na základě nahraných dat má 1 280 uzlů a 1 294 hran.



Obrázek 16 Graf zobrazující Kalifornskou síť silnici na základě dat získaných z Internetu (Zdroj vlastní.)

Pro zjištění nejkratší cesty je použita funkce `algo.shortestPath.stream()` pomocí následujícího dotazu, za předpokladu, že startovní uzel má ID „915“ a cílový uzel má ID „1153“:

```
MATCH (start:Crossroad {id: "915"}), (end:Crossroad {id: "1153"})
CALL algo.shortestPath.stream(start, end, "distance") YIELD nodeId,
cost
RETURN algo.getNodeById(nodeId).id AS ID, cost AS Distance
```

Pomocí algoritmu byla zjištěna nejkratší cesta (obrázek 17) z křižovatky ID „915“ do křižovatky ID „1153“, má celkem 11 uzlů s celkovou vzdáleností 0,2115.

ID	Distance
"915"	0.0
"916"	0.021244
"913"	0.036413
"1027"	0.057432
"1028"	0.06688
"1029"	0.16999199999999998
"1030"	0.17250699999999997
"1031"	0.18687499999999996
"1026"	0.19105399999999995
"1032"	0.20033699999999993
"1153"	0.21148999999999993

Obrázek 17 Tabulka zobrazující nejkratší cesty u určitého uzlu na datech Kalifornské sítě silnic (Zdroj vlastní.)

Dalším grafovým algoritmem v kontextu hledání cest je *Random Walk algorithm* (v češtině Náhodná procházka). Algoritmus poskytuje náhodné cesty v grafu. Náhodná procházka k jednomu z uzlů je náhodná procházka k jednomu z uzlů ("6.7. The Random Walk algorithm - Chapter 6. Path finding algorithms," n.d.).

Pro znázornění algoritmu náhodná procházka jsou využita stejná data (Kalifornské sítě silnic) za použití funkce `algo.randomWalk.stream()`. Pro zjištění čtyř cest za pomocí dvou kroků z uzlu s ID „911" je dotaz následující:

```
MATCH (c:Crossroad {id: "911"})
CALL algo.randomWalk.stream(id(c), 2, 4, {path:true})
YIELD nodeIds
UNWIND nodeIds AS nodeId
RETURN algo.getNodeById(nodeId).id AS Step
```

Tabulka (obrázek 18) zobrazuje výčet (celkem 12 záznamů) provedených kroků na základě dotazu, kde sloupec Step značí ID navštívených křižovatek.

Step
"911"
"910"
"909"
"911"
"912"
"911"
"911"
"910"
"911"
"911"
"912"
"890"

Obrázek 18 Tabulka zobrazující náhodou procházku určitého uzlu na Kalifornské síti silnic (Zdroj vlastní.)

5.3.3 Predikce propojení

Predikce propojení nachází své uplatnění v problematice sociálních sítí či jakýkoliv typ doporučujících systémů. Sociální sítě se často skládají z chybějících či dokonce falešných vazeb a předvídání těchto vazeb pomůže lépe porozumět jejich mechanismům. Predikce propojení se obecně pokouší porozumět asociaci mezi dvěma uzly (Jalili Mahdi, Orouskhani Yasin, Asgari Milad, Alipourfard Nazanin, & Perc Matjaž, 2017).

Jedním z příkladů algoritmů založený na principu predikce propojení je *Common Neighbors algorithm* (v češtině Algoritmus společných sousedů). Algoritmus zachycuje myšlenku, že dva uzly, které mají společného „přítele“, mají větší šanci se seznámit, než ty uzly, které nemají žádného společného „přítele“ (“8.2. The Common Neighbors algorithm - Chapter 8. Link Prediction algorithms,” n.d.). Své uplatnění nachází především v problematice sociálních sítí.

Pro znázornění algoritmu společných sousedů jsou využita data ze sociálních sítí Twitter (použitá výše) a to následovně za pomocí dotazu:

```
MATCH (u1:User)--(u2:User)--(u3:User)
WHERE NOT (u1) = (u3)
OPTIONAL MATCH (u1)-[f:FOLLOWS]->(u3)
RETURN u1.id, u3.id, count(u2) AS Count,
ORDER BY Count DESC
```

Hledáme společné sousedy, tedy všechny uzly `u2`, uzlů `u1` a `u3`. Z tabulky (obrázek 19) je patrné, že nejvíce společných sousedů mají mezi sebou uzly s ID „6088382“ a „8630562“, kde celkový počet celkových společných sousedů je 69.

u1.id	u3.id	Count
"6088382"	"8630562"	69

Obrázek 19 Tabulka zobrazující největší počet společných sousedů na základě dvou určitých uzlů z dat sociální sítě Twitter (Zdroj vlastní.)

6 Analýza dat v rámci Neo4j

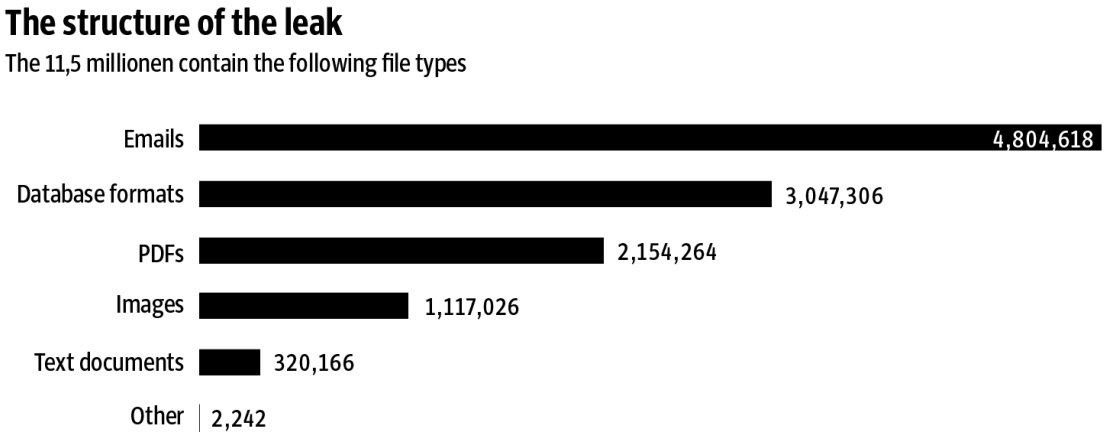
6.1 Úvod do analýzy

Prováděná analýza dat je v rámci grafové databáze Neo4j za pomocí instalovaného Neo4j Serveru s jeho vizualizačním prostředí Neo4j Browser a dotazovacího jazyka Cypher. Analyzovaná data se týkají problematiky Panama Papers, která lze získat volně z Internetu několika způsoby.

Nad daty budou prováděny dotazy i za použití grafových algoritmů v rámci dataminingu. Postup dataminingu dat pomocí dotazů a jejich vrácené hodnoty budou následně interpretovány.

6.2 Problematika Panama Papers

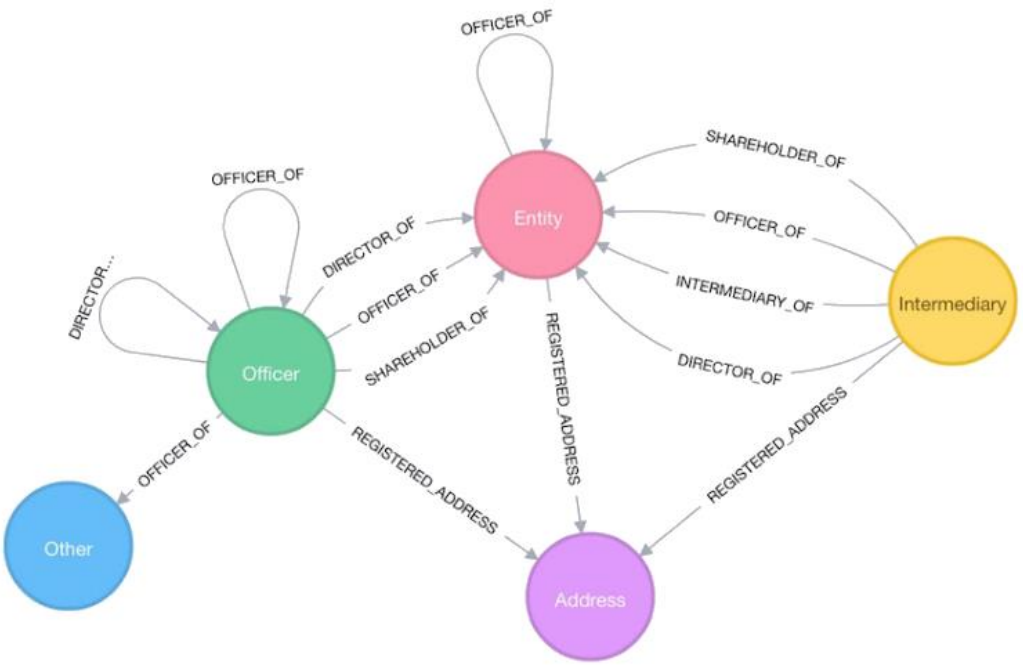
Pojem Panama Papers odkazuje na přibližně 11,5 milionu dokumentů pokrývající období od 70. let do začátku roku 2016, především e-maily, soubory PDF, fotografie apod. (obrázek 20), které byly anonymním zdrojem předloženy německým novinám Süddeutsche Zeitung (Obermaier, Obermayer, Wormer, & Jaschensky, n.d.). Dokumenty byly majetkem panamské advokátní kanceláře Mossack Fonseca a odhalily síť více než 214 tisíc daňových rájů zahrnující osoby a subjekty z 200 různých zemí (Kenton, 2018). Než byly dokumenty zveřejněny pro širokou veřejnost, The International Consortium of Investigative Journalists (ICIJ) se podílelo na jejich analýze s použitím grafové databáze Neo4j.



Obrázek 20 Struktura uniklých dat Panama Papers (Obermaier et al., n.d.)

6.3 Data a import dat

Základní datový model pro Panama Papers (obrázek 21) se skládá z pěti uzlů (nodes) a pěti hran (relationships).



Obrázek 21 Datový model pro Panama Papers (Lyon, 2018)

Pět základních uzlů datového modelu pro Panama Papers:

- Entity – offshorová právnická osoba[5]. Může se jednat o společnost, nadaci či jinou právnickou osobu,
- Officer – osoba nebo společnost, která hraje roli v offshorovém subjektu, jako je příjemce, ředitel nebo akcionář,
- Intermediary – zprostředkovatel (obvykle právnická firma nebo prostředník), která žádá offshorové poskytovatele služeb vytvořit offshorovou firmu,
- Address – registrovaná adresa, jak je uvedena v původních databázích získaných ICIJ,
- Other – ostatní subjekty nalezené v datech.

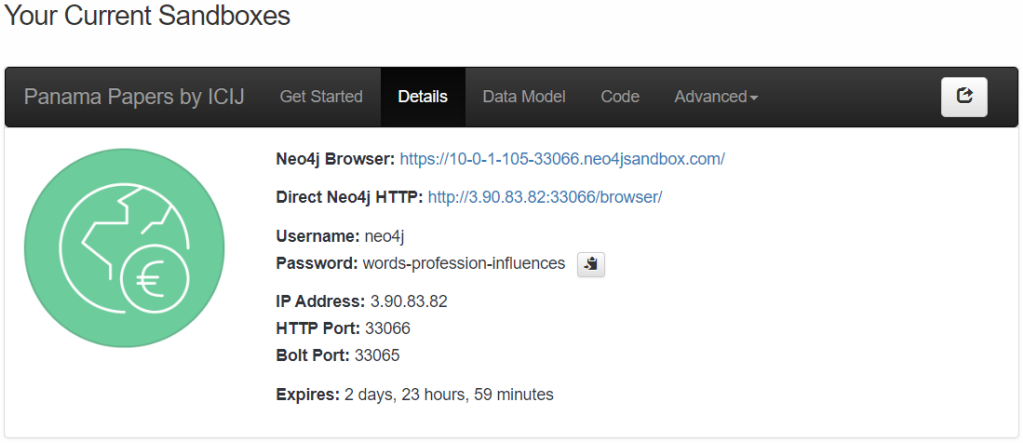
[5] společnost registrovaná v zemích umožňující příznivější daňový režim

Dále datový model (obrázek 21) zobrazuje také pět základních hran – vztahů (relationships):

- OFFICER_OF – příjemce daného uzlu,
- SHAREHOLDER_OF – akcionář daného uzlu,
- DIRECTOR_OF – ředitel daného uzlu,
- INTERMEDIARY_OF – zprostředkovatel daného uzlu,
- REGISTERED_ADDRESS – registrovaná adresa daného uzlu.

Získat data k Panama Papers lze dvěma způsoby. V případě importu dat pomocí Console a dotazovacího jazyka Cypher lze získat data z oficiálních webových stránek Offshore Leaks Database by ICIJ ("How to download this database | ICIJ Offshore Leaks Database," n.d.). Databáze ICIJ Offshore Leaks je licencovaná pod licencí Open Database License a její obsah pod licencí Creative Commons Attribution-ShareAlike. Data pro Panama Papers jsou ve formátu CSV souborů. CSV souborů je celkem 5 (4 soubory pro uzly a 1 soubor pro vztahy) a jejich celková velikost je přibližně 174 MB. Import probíhá použitím dotazu `LOAD CSV`.

Druhý způsob, jak získat data, je použití Neo4j Sandboxu. Neo4j Sandbox nabízí uživateli možnost stáhnout si jeden z několika možných populárních příkladů použití, mezi nimi jsou i Panama Papers by ICIJ. Po kliknutí na *Launch Sandbox* je uživateli zpřístupněná instance s nahranou datovou sadou (obrázek 22). Přes `http` odkaz je možné s touto instancí pracovat přes Neo4j Browser.



Obrázek 22 Neo4j Sandbox s Panama Papers instancí (Zdroj vlastní.)

V rámci analýzy bakalářské práce bude využit způsob přes Neo4j Sandbox a to z několika hlavních důvodů. Panama Papers Sandbox poskytuje mnohem více dat, než výše zmíněné

CSV soubory, s mnohem více různými typy hran – vztahů. Více možných typů vztahů umožní detailnější analýzu a její interpretaci. Dále při bližší inspekci CSV souborů bylo zjištěno mnoho chyb v jednotlivých záznamech – např. různé použití uvozovacích znaků, neoddělení jednotlivých záznamů apod.

Celková velikost databáze (obrázek 23) je téměř 942 MiB. Obrázek dále například zobrazuje velikost celkových uzlů, hran, vlastností apod.

Store Sizes

Count Store	57.94 KiB
Label Store	8.00 MiB
Index Store	624.00 KiB
Schema Store	4.00 MiB
Array Store	4.00 MiB
Logical Log	88 B
Node Store	20.00 MiB
Property Store	312.00 MiB
Relationship Store	64.00 MiB
String Store	388.00 MiB
Total Store Size	941.79 MiB

Obrázek 23 Velikost databáze Panama Papers (Zdroj vlastní.)

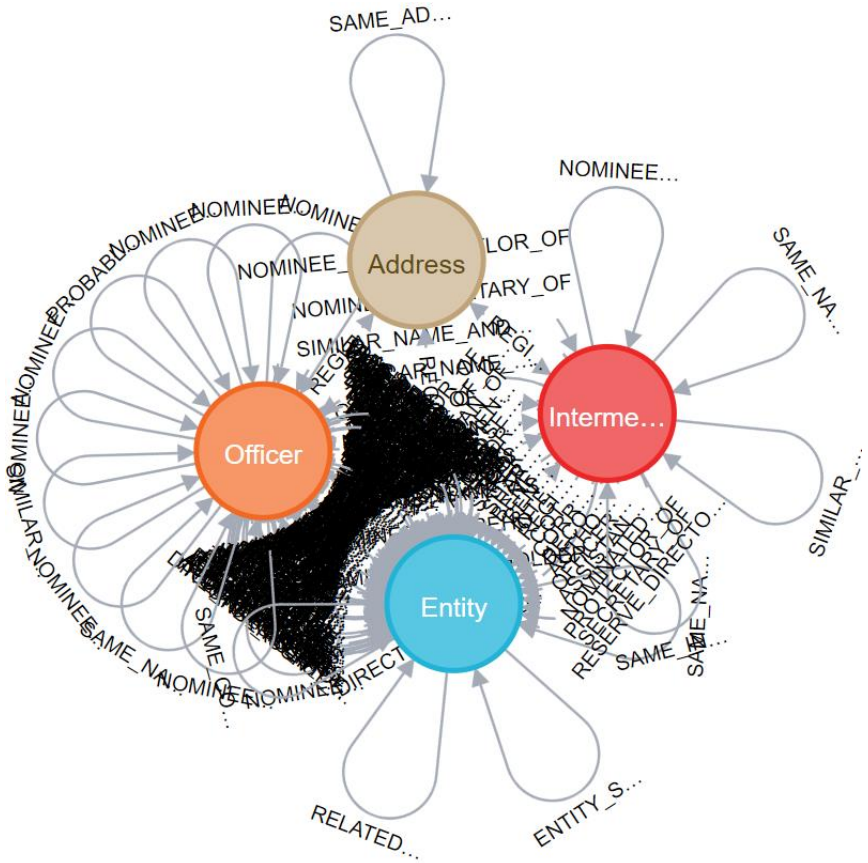
Po provedení dotazů `COUNT` pomocí dotazovacího jazyka Cypher bylo zjištěno, že Panama Papers databáze celkem obsahuje 1 040 535 uzlů a 1 535 552 hran. Veškeré typy uzlů, hran a vlastností lze zobrazit v rámci Neo4j Browseru na postranním panelu v záložce *Informace o databázi (Database Information)*.

6.4 Analýza Panama Papers data setu

Před detailní analýzou je důležité si uvědomit s jakými daty se pracuje. Základní datový model (obrázek 21) zmíněný výše, popisuje základní uzly a hrany vyskytující se v databázi. Pro zobrazení skutečného datového modelu (či schématu) použijeme dotazující příkaz `CALL`, který se používá k volání procedury zavedené v databázi. Dotaz zobrazující datový model databáze:

```
CALL db.schema();
```

Z datového modelu (obrázek 24) je patrné, že mezi uzly (Address, Officer, Intermediary a Entity) existuje mnoho hran, kde není zcela možné je zobrazit na malém místě.



Obrázek 24 Detailní datový model Panama Papers použitý v Neo4j SandBoxu (Zdroj vlastní.)

I přesto, že *Informace o databázi (Database Information)* sděluje, jaké množství uzlů se v databázi nachází, uživatel nemá dostatečné informace, jaký počet mají jednotlivé typy uzlů. Informaci získáme pomocí následujícího dotazu:

```
MATCH (node)
RETURN labels(node) AS type,count(*)
ORDER BY count(*) DESC
```

Výsledná tabulka (obrázek 25) je podle příkazu rozdělena na dva sloupce, kde první sloupec určuje o jaký typ uzlu se jedná, a druhý sloupec představuje počet uzlů pro jednotlivý typ. Hodnoty jsou seřazeny sestupně. Z tabulky vyplývá, že nejvíce uzlů patří do typu Entity a nejméně uzlů do typy Intermediary.

type	count(*)
["Entity"]	495038
["Officer"]	369715
["Address"]	151605
["Intermediary"]	24177

Obrázek 25 Tabulka zobrazující počet uzlů pro jednotlivé typy uzlů (Zdroj vlastní.)

Další zajímavou informací ohledně problematiky Panama Papers (v rámci data setu[6]) může představovat jaký typ uzlu (Entity, Officer a Intermediary) mají nejvíce hran k určitému typu uzlu. Pokud bychom se zajímali o typ Intermediary a jeho směřování k typu Entity, dotaz by byl následující:

```
MATCH (i:Intermediary), (e:Entity)
MATCH (i)-[connection]-(e)
RETURN i.name AS Intermediary, type(connection) As relationship,
head(labels(e)) AS type, count(*) AS count
ORDER BY count DESC LIMIT 5
```

Tabulka (obrázek 26) zobrazuje prvních pět výsledků s nejvyšším počtem hran uzlů typu Intermediary. Například první záznam v tabulce udává, že uzel s názvem *"Portcullis TrustNet (BVI) Limited"* patřící do typu uzlů Intermediary, má nejvíce hran typu *"RECORDS_REGISTERS_OF"*, které směřují do uzlu typu Entity. Počet hran je 36 238.

Intermediary	relationship	type	count
"Portcullis TrustNet (BVI) Limited"	"RECORDS_REGISTERS_OF"	"Entity"	36238
"MOSSACK FONSECA & CO. (BAHAMAS) LIMITED"	"INTERMEDIARY_OF"	"Entity"	14901
"UBS TRUSTEES (BAHAMAS) LTD."	"INTERMEDIARY_OF"	"Entity"	9717
"CREDIT SUISSE TRUST LIMITED"	"INTERMEDIARY_OF"	"Entity"	8299
"TRIDENT CORPORATE SERVICES (BAH) LTD"	"INTERMEDIARY_OF"	"Entity"	8286

Obrázek 26 Tabulka zobrazující nejvyšší počet hran uzlu Intermediary (Zdroj vlastní.)

[6] kolekce dat, konkrétně Panama Papers data

Za pomocí obdobného příkazu s použitím na typ uzlu Officer získáme podobnou tabulku (obrázek 27). Z tabulky je patrné, že v prvním záznamu figuruje opět *"Portcullis TrustNet (Samoa) Limited"*.

Officer	relationship	type	count
"Portcullis TrustNet (Samoa) Limited"	"SHAREHOLDER_OF"	"Entity"	4227
"MOSSFON SUBSCRIBERS LTD."	"SHAREHOLDER_OF"	"Entity"	3882
"Standard Directors Ltd."	"DIRECTOR_OF"	"Entity"	1992
"Execorp Limited"	"DIRECTOR_OF"	"Entity"	1847
"Sharecorp Limited"	"SHAREHOLDER_OF"	"Entity"	1610

Obrázek 27 Tabulka zobrazující nejvyšší počet hran uzlu Officer (Zdroj vlastní.)

Opět s použitím obdobného příkazu získáme další tabulku (obrázek 28) ohledně typu uzlu Entity, se směřování k uzlu například Officer. První záznam v tabulce udává, že uzel typu Entity s názvem *"ACCELONIC LTD."* má nejvíce hran s názvem *"SHAREHOLDER_OF"*, přičemž počet hran je 1 006.

Entity	relationship	type	count
"ACCELONIC LTD."	"SHAREHOLDER_OF"	"Officer"	1006
"Dale Capital Group Limited"	"SHAREHOLDER_OF"	"Officer"	504
"VELA GAS INVESTMENTS LTD."	"SHAREHOLDER_OF"	"Officer"	492
"WAN CHI INVESTMENTS LIMITED"	"SHAREHOLDER_OF"	"Officer"	447
"HANNSPREE INC."	"SHAREHOLDER_OF"	"Officer"	430

Obrázek 28 Tabulka zobrazující nejvyšší počet hran uzlu Entity (Zdroj vlastní.)

Pokud bychom se zajímali o více konkrétnější informace, mohlo by být podstatné, jaké země jsou uvedeny v data setu ohledně Panama Papers. K takové informaci je možné se dostat přes uzly typu Address. Každý uzel typu Address v sobě uchovává vlastnosti, kterým lze přistupovat. Vlastnost figurující v následujícím příkazu je *countries*, podle které lze zjistit, o jaké adresy v data setu se jedná. Dotaz je následující:

```
MATCH (a:Address) WHERE exists(a.countries)
RETURN a.countries AS country, count(*)
ORDER BY count(*) DESC
```

Výše zmíněný příkaz vrátil tabulku, která obsahuje 209 záznamů, tedy 209 zemí figurující v Panama Papers v rámci data setu.

Tabulka (obrázek 29) zobrazuje prvních pět nejčastějších zemí, kde je počet výskytů seřazen sestupně:

country	count(*)
"China"	28073
"Hong Kong"	21041
"Taiwan"	14610
"United States"	6861
"Singapore"	5728

Obrázek 29 Tabulka nejčastěji vyskytujících se zemí v rámci uzlů typu Address (Zdroj vlastní.)

Jak je z tabulky (obrázek 29) vidět, nejčastěji figurující země (28 073 výskytů), která se vyskytuje v data setu je Čína. Mezi 209 zeměmi se vyskytuje i Česká republika.

Se soustředěním na Českou republiku lze zjistit, jaké osoby (Officer) s adresou v České republice mají nějaký vztah s určitými společnostmi (Entity). To lze zjistit pomocí následujícího dotazu:

```
MATCH (o:Officer)--(e:Entity)
WHERE o.country_codes = "CZE"
RETURN o.address AS Address, o.name AS Officer, e.name AS Entity
```

Výše zmíněný dotaz vrátil 303 záznamů. Tabulka (obrázek 30) zobrazuje prvních pět záznamů. I z těchto pěti záznamů je patrné, že osoby mající nějaký vztah k určité společnosti jsou z různých měst po celé České republice. Jisté adresy se vyskytují i vícekrát na jiné jméno. (Je také patrné, že některé adresy nejsou korektně napsané.)

Address	Officer	Entity
"1 Melnicka Street; 150 00; Praha 5; Czech Republic"	"JAKUB SVOBODA"	"T.I.S. Investments Ltd."
"1321 Nam Svobody; 75501 Vsetin; Czech Republic"	"TOMAS QUIS"	"GENERAL TRADING & CONSULTING CORP."
"29/5 NAKOCOURKACH; PRAGUE 6; 16900; CZECH REPUBLIC"	"PETR NOSCAK"	"DURALMA LIMITED"
"29/5 Nakocourkoch; Praque 6 16900; Czech Republic"	"Branston Haggerty"	"ENSWELL LTD."
"5 LAD. KOUBKA; KARLOVY VARY; 36001; CZECH REPUBLIC"	"Alexander VASILKOV"	"METCALFE ENGINEERING CORP."

Obrázek 30 Tabulka zobrazující české adresy osob mající vztah k určitým společnostem (Zdroj vlastní.)

Po bližším zkoumání osob v tabulce (tabulka 30) s 303 záznamy se vyskytují osoby nejen s českými jmény, ale dále se vyskytují osoby s podobným příjmením či jméno osoby je uvedeno jako *„THE BEARER“*[7]. Jedná se o akcii na doručitele či akcii na majitele, kdy k akcii není přiřazeno žádné jméno z hlediska veřejných rejstříků, tudíž není známý její vlastník (“Listinné akcie na doručitele / NFPK,” n.d.).

Pokud bychom chtěli zjistit, kolik záznamů je na neznámé osoby pod jménem *„THE BEARER“*, použije se následovný dotaz:

```
MATCH (o:Officer)--(e:Entity)
WHERE o.country_codes = "CZE" AND o.name CONTAINS "BEARER"
MATCH (o:Officer)-[r]-(e:Entity)
RETURN o.address AS Address, o.name AS Officer,
type(r) AS relationship, e.name AS Entity
ORDER BY o.address
```

Z tabulky (obrázek 31), kde je pro ilustraci zobrazeno pouze prvních pět záznamů (celkových záznamů je 52) , lze vyčíst, že anonymní osoby jsou akcionáři určitých společností (Entity).

Address	Officer	relationship	Entity
"Bellusov 1802/3, Prague 5 Stodulky"	"THE BEARER"	"SHAREHOLDER_OF"	"DAMON INVEST LIMITED"
"Bellusov 1802/3, Prague 5 Stodulky"	"THE BEARER"	"SHAREHOLDER_OF"	"DAMON INVEST LIMITED"
"Elisky Krasnohorske 11. 110 00 Praha 1,Czech Republic"	"THE BEARER"	"SHAREHOLDER_OF"	"St. Catherine´s Institute for Economic Research and Education Ltd."
"Elisky Krasnohorske 11. 110 00 Praha 1,Czech Republic"	"THE BEARER"	"SHAREHOLDER_OF"	"St. Catherine´s Institute for Economic Research and Education Ltd."
"FRANCOUZSKA 74, PRAHA 10 10100 CZECH REPUBLIC"	"THE BEARER"	"SHAREHOLDER_OF"	"TELFA IMPEX LIMITED"

Obrázek 31 Tabulka zobrazující detaily o osobách pod jménem "THE BEARER" v České republice (Zdroj vlastní.)

Z tabulky je také patrné, že na určitou adresu je několik anonymních záznamů. Pokud bychom vyfiltrovali opakující se záznamy, zjistili bychom, že z 52 celkových adres se jedná pouze o 16 jedinečných adres.

[7] v češtině doručitel

Pokud se zaměříme na společnosti, ke kterým mají anonymní osoby vztah *„SHAREHOLDER_OF“*, zjistíme, z jakých zemí společnosti jsou a jak často se v data setu vyskytují v kontextu anonymních akcionářů v České republice, a to pomocí dotazu:

```
MATCH (o:Officer)--(e:Entity)
WHERE o.country_codes = "CZE" AND o.name CONTAINS "BEARER"
MATCH (o:Officer)-[r]-(e:Entity)
RETURN e.name AS Entity, e.countries AS Country, count(*)
ORDER BY count(*) DESC
```

Výše zmíněný dotaz vrátil tabulku (obrázek 32), která se skládá z tří sloupců, kde první sloupec udává název společnosti, druhý sloupec udává zemi, kterou má společnost uvedenou v adrese a třetí sloupec zobrazuje výčet výskytů této společnosti v rámci anonymních akcionářů v České republice. Tabulka naznačuje, se jedná pouze o patnáct jedinečných společností, které jsou celkem registrované pouze do 5 pěti zemí (Monako, Rusko, Samoa, Seychelly a Velká Británie), kde značnou převahu má země Samoa.

Entity	Country	count(*)
"CG INVEST LTD."	"Samoa"	16
"SPEEDYSHARE LTD."	"Samoa"	10
"MELBOURNE HOUSE LTD."	"Monaco"	4
"ALBUCON LIMITED"	"Samoa"	3
"NOVIA TRADING LTD."	"Samoa"	2
"SAXTONIA ENTERPRISES LTD."	"Samoa"	2
"ANETHUM CORP."	"Seychelles"	2
"ELEON LTD."	"Samoa"	2
"TELECOM WORLDWIDE LTD."	"Samoa"	2
"St. Catherine´s Institute for Economic Research and Education Ltd."	"Seychelles"	2
"DAMON INVEST LIMITED"	"Samoa"	2
"ORP CAPITAL LTD."	"Seychelles"	2
"SILVERDALE LIMITED"	"United Kingdom"	1
"TELFA IMPEX LIMITED"	"Russia"	1
"M.S.F. LIMITED"	"Seychelles"	1

Obrázek 32 Tabulka zobrazující zastoupení společností, kde akcionáři jsou anonymní osoby z České republiky (Zdroj vlastní.)

Po bližší analýze byl zjištěný vztah mezi uzly typu Entity a typem Intermediary v kontextu České republiky, kde akcionáři vystupují pod jménem „THE BEARER“, a to *„INTERMEDIARY_OF“*. Získání takových údajů zajišťuje dotaz:

```
MATCH (o:Officer)--(e:Entity)--(i:Intermediary)
WHERE o.country_codes = "CZE" AND o.name CONTAINS "BEARER"
MATCH (e:Entity)-[r]-(i:Intermediary)
RETURN DISTINCT i.name AS Intermediary, type(r) AS Relationship,
e.name AS Entity
ORDER BY i.name
```

Tabulka (obrázek 33) zobrazuje výčet společností, které jsou určitými zprostředkovateli ke společnostem s ručením omezeným zmíněné výše v rámci analýzy. V tabulce se nachází 5 unikátních společností, kde nejčastěji vyskytující se společnost je *„MOSSACK FONSECA & CO. CZ, S.R.O“*.

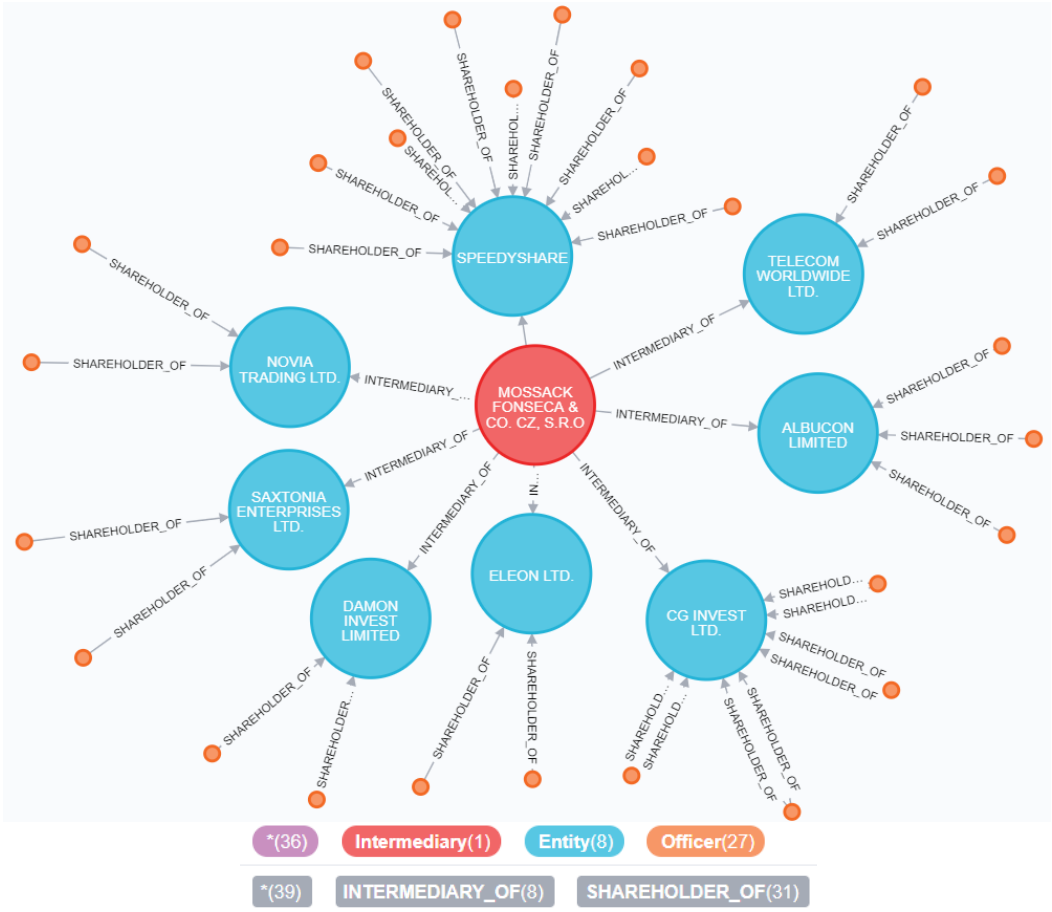
Intermediary	Relationship	Entity
"CORPORATE BUSINESS CENT"	"INTERMEDIARY_OF"	"SILVERDALE LIMITED"
"LEGAL CONSULTING SERVICES LIMITED"	"INTERMEDIARY_OF"	"TELFA IMPEX LIMITED"
"MOORES ROWLAND CORPORATE SERVICES"	"INTERMEDIARY_OF"	"MELBOURNE HOUSE LTD."
"MOSSACK FONSECA & CO. CZ, S.R.O"	"INTERMEDIARY_OF"	"CG INVEST LTD."
"MOSSACK FONSECA & CO. CZ, S.R.O"	"INTERMEDIARY_OF"	"SAXTONIA ENTERPRISES LTD."
"MOSSACK FONSECA & CO. CZ, S.R.O"	"INTERMEDIARY_OF"	"DAMON INVEST LIMITED"
"MOSSACK FONSECA & CO. CZ, S.R.O"	"INTERMEDIARY_OF"	"SPEEDYSHARE LTD."
"MOSSACK FONSECA & CO. CZ, S.R.O"	"INTERMEDIARY_OF"	"ALBUCON LIMITED"
"MOSSACK FONSECA & CO. CZ, S.R.O"	"INTERMEDIARY_OF"	"NOVIA TRADING LTD."
"MOSSACK FONSECA & CO. CZ, S.R.O"	"INTERMEDIARY_OF"	"TELECOM WORLDWIDE LTD."
"MOSSACK FONSECA & CO. CZ, S.R.O"	"INTERMEDIARY_OF"	"ELEON LTD."
"WABERIA CONSULTING LLC"	"INTERMEDIARY_OF"	"St. Catherine´s Institute for Economic Research and Education Ltd."
"WABERIA CONSULTING LLC"	"INTERMEDIARY_OF"	"ANETHUM CORP."
"WABERIA CONSULTING LLC"	"INTERMEDIARY_OF"	"ORP CAPITAL LTD."
"WABERIA CONSULTING LLC"	"INTERMEDIARY_OF"	"M.S.F. LIMITED"

Obrázek 33 Tabulka zobrazující společnosti, které jsou zprostředkovateli k analyzovaným společnostem v rámci anonymních akcionářů s adresou v České republice (Zdroj vlastní.)

Vzhledem k tomu, že společnost *„MOSSACK FONSECA & CO CZ, S.R.O.“* se v tabulce vyskytuje celkem osmkrát, lze vztahy mezi společnostmi typu Intermediary a Entity znázornit i graficky, a to následovně:

```
MATCH (o:Officer)--(e:Entity)--(i:Intermediary)
WHERE o.country_codes = "CZE" AND o.name CONTAINS "BEARER" AND i.name
CONTAINS "MOSSACK"
MATCH (e:Entity)-[r]-(i:Intermediary)
RETURN DISTINCT i, e, o, r
```

Výsledný graf (obrázek 34) představuje grafické znázornění splečnosti *„MOSSACK FONSECA & CO CZ, S.R.O.“* a její vztahy mezi ostatními společnostmi, kde akcionáři těchto společností jsou anonymní osoby z České republiky. Graf zobrazuje celkem 39 uzlů a 39 hran.



Obrázek 34 Graf zobrazující společnost „MOSSACK FONSECA & CO. CZ S.R.O.“ a její vztahy mezi ostatními společnostmi spolu s českými anonymními akcionáři (Zdroj vlastní.)

Pokud se zaměříme na společnosti, které mají ve jméně *„MOSSACK FONSECA & CO.“* jednoduchým dotazem:

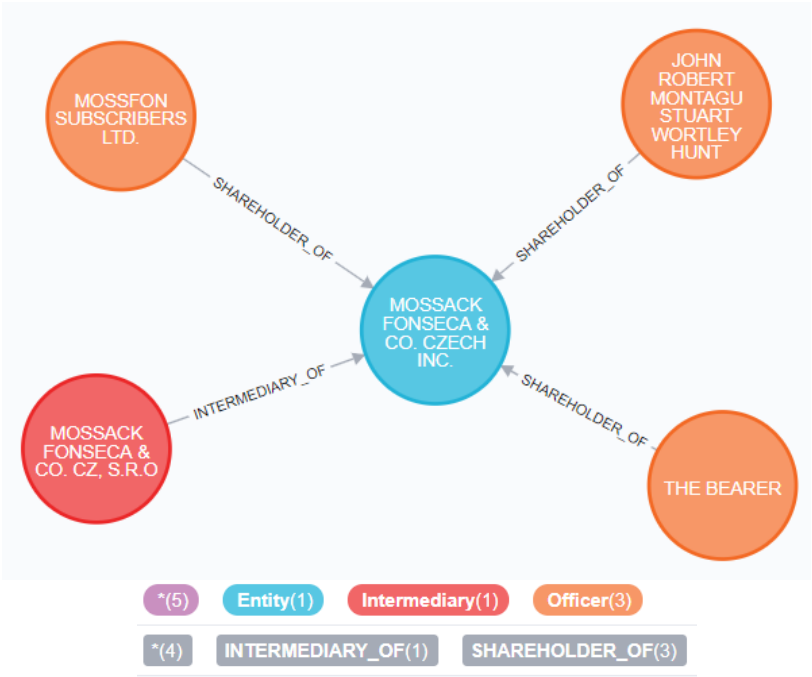
```
MATCH (n)
WHERE n.name CONTAINS "MOSSACK FONSECA & CO."
RETURN n.name
```

Zjistíme, že společnosti obsahující výše dotazovaný název v data setu Panama Papers má 296 záznamů. Při bližším zkoumání, jsou některé záznamy identické, pouze obsahují gramatické chyby v názvu apod. Zajímavý byl však výskyt společnosti *„MOSSACK FONSECA & CO. CZECH INC.“* a společnosti *„MOSSACK FONSECA & CO. CZ, S.R.O.“* V tomto případě se však nejedná o identické společnosti.

Pro zobrazení výše zmíněných dvou společností použijeme dotaz a po vykreslení v Neoj4 Browseru u uzlu typu Entity rozklikneme zobrazení potomků:

```
MATCH (n)
WHERE n.name CONTAINS "MOSSACK FONSECA & CO. CZECH INC."
RETURN n
```

Graf (obrázek 35) zobrazuje společnost „*MOSSACK FONSECA & CO. CZECH INC.*“ a uzly, které k ní mají nějaké vztahy.

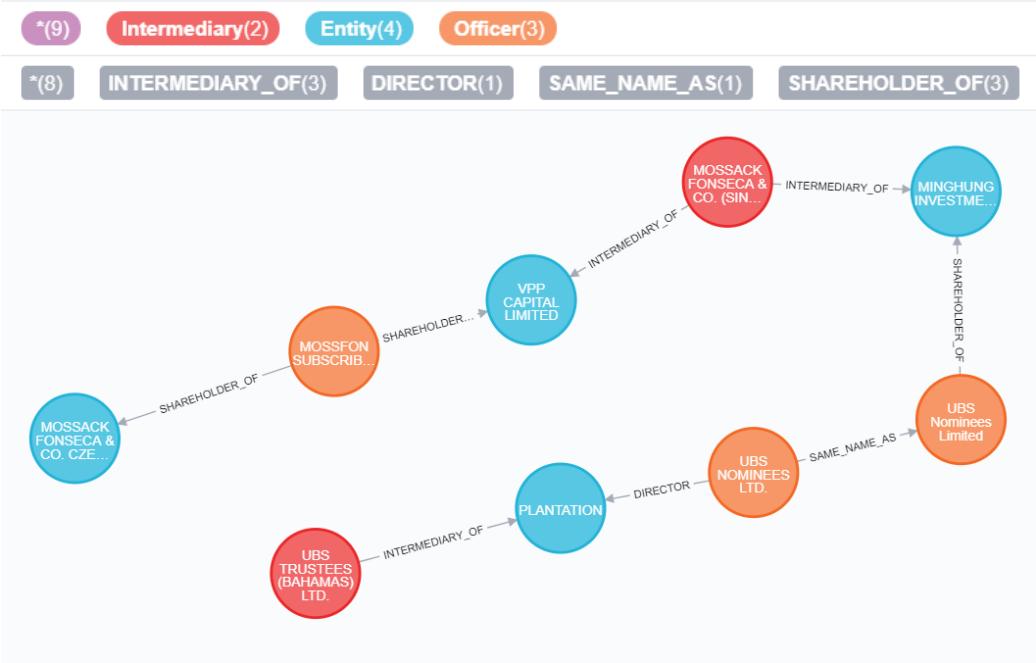


Obrázek 35 Graf zobrazující společnosti „MOSSACK FONSECA & CO. CZECH INC.“ a „MOSSACK FONSECA & CO. CZ, S.R.O.“ (Zdroj vlastní.)

Analyzovat Panama Papers data set lze i pomocí grafových algoritmů v rámci instalovaného pluginu Neo4j. Pro zjištění nejkratší cesty pomocí Dijkstrova algoritmu, v rámci Cypheru funkce `algo.shortestPath.stream()` například ze společnosti *„UBS TRUSTEES (BAHAMAS) LTD.“*, která se vyskytuje v tabulce (obrázek 26), do společnosti *„MOSSACK FONSECA & CO. CZECH INC.“* je dotaz následující:

```
MATCH (start:Intermediary {name: "UBS TRUSTEES (BAHAMAS) LTD."}),
(end:Entity {name: "MOSSACK FONSECA & CO. CZECH INC."})
CALL algo.shortestPath.stream(start, end, "", {direction: "BOTH"})
YIELD nodeId
RETURN algo.getNodeById(nodeId) AS ID
```

Výše zmíněný dotaz vrací graf (obrázek 36), který se skládá z 9 uzlů a 8 hran. Nejkratší cesta prochází třemi typy uzlů, přičemž dva uzly jsou si navzájem podobné (mají stejné jméno) – společnost *„UBS NOMINEES LTD.“* se společností *„UBS Nominees Limited“*. Na základě tohoto poznatku a skutečností, že existuje hrana s jménem „SAME_NAME_AS“, je zřejmé, že se v data setu vyskytují určité uzly opakovaně, i přesto, že každý uzel má své jedinečné ID.



Obrázek 36 Graf zobrazují nejkratší cestu ze společnosti „UBS TRUSTEES (BAHAMAS) LTD.“ do společnosti „MOSSACK FONSECA & CO. CZECH INC.“ (Zdroj vlastní.)

Z předešlého obrázku (obrázek 36) je patrné, že se v data setu vyskytují uzly, které mají stejné jméno, ale přesto jsou zaznamenány odděleně. Pokud bychom chtěli zjistit počet takových uzlů, je třeba se soustředit na hrany neboli vztahy, jež určují tuto duplicitu, a to je hrana s názvem *„SAME_NAME_AS“*. Jelikož data set má široké množství hran, není přesně jasné, zda se nevyskytují další jiné hrany zaznamenávající nějakou jinou duplicitu. Proto je použit dotaz, pro zjištění hran, jejichž jméno (či typ) obsahuje slovo *„SAME“*, pokud ano, dotaz vrátí i počet výskytů takových typů uzlů:

```
MATCH ()-[r]->() WHERE type(r) CONTAINS "SAME"
RETURN type(r) AS name, count(*) AS count
ORDER BY count(*) DESC
```

Výsledná tabulka (obrázek 37) vrací 6 takových záznamů duplicit, kde právě nejčastější výskyt má hrana s názvem *„SAME_NAME_AS“* a to konkrétně 20 561 výskytů. Co však může být zajímavé je hrana s názvem *„PROBABLY_SAME_OFFCER_AS“* vyskutující se celkem 132, což může znamenat nějakou spekulaci v podobnosti názvu uzlů typů Officer.

name	count
"SAME_NAME_AS"	20429
"SAME_COMPANY_AS"	15523
"SAME_NAME_AND_REGISTRATION_DATE_AS"	3146
"SAME_ADDRESS_AS"	965
"PROBABLY_SAME_OFFICER_AS"	132
"SAME_INTERMEDIARY_AS"	4

Obrázek 37 Tabulka zobrazující výskyt hran, jejichž jméno obsahuje slovo „SAME“ (Zdroj vlastní.)

Pokud bychom se zajímali o jakou podobnost se pravděpodobně jedná, zobrazíme daný typ hrany pomocí dotazu:

```
MATCH (o1:Officer)-[r:PROBABLY_SAME_OFFICER_AS]->(o2:Officer)
RETURN o1.name, o2.name
LIMIT 7
```

Z tabulky (obrázek 38) lze vyčíst, že jistá podobnost je postavena pouze na jinak zapsaných jmen uzlů typu Officer, což znamená rozdílnost v malých a velkých písmenech, výskyt interpunkce, změna pořadí slov apod., tudíž tento typ hrany může být nahrazen za *„SAME_NAME_AS“*, pokud by to bylo nutné k další analýze.

o1.name	o2.name
"ROBERT JOSEPH ANDRES"	"ANDRES ROBERT JOSEPH"
"LIMITED BASSIM ENTERPRISES"	"Bassim Enterprises Limited"
"TANG HOI LAM TONY"	"TONY TANG HOI LAM"
"WINDSOR PARK CORPORATION S.A."	"WINDSOR PARK CORPORATION, S.A."
"Clementi Limited"	"CLEMENTI LIMITED"
"ANDREAS SVARD LARS RICHARD"	"MR. LARS RICHARD ANDREAS SVARD"
"REDMOUNT NOMINEES LIMITED"	"LIMITED REDMOUNT NOMINEES"

Obrázek 38 Tabulka zobrazující uzly spojené hranou „PROBABLY_SAME_OFFICER_AS“ (Zdroj vlastní.)

V rámci dotazovacího jazyka Cypher nelze přejmenovat hranu na jiný název přímo. Pokud však trváme na změně jména, je třeba nejprve odstranit stávající typ hran a vytvořit typ jiný, a to následovně:

```
MATCH (n1)-[old_r:PROBABLY_SAME_OFFICER_AS]->(n2)
CREATE (n1)-[new_r:SAME_NAME_AS]->(n2)
DELETE old_r
```

Výsledná tabulka (obrázek 39) s porovnáním s tabulkou výše zmíněnou (obrázek 37) zobrazuje, jaké změny proběhly po přejmenování hran. Je znatelné, že k hraně jménem *„SAME_NAME_AS“* přibylo 132 hran z eliminované hrany *„PROBABLY_SAME_OFFICER_AS“*.

name	count
"SAME_NAME_AS"	20561
"SAME_COMPANY_AS"	15523
"SAME_NAME_AND_REGISTRATION_DATE_AS"	3146
"SAME_ADDRESS_AS"	965
"SAME_INTERMEDIARY_AS"	4

Obrázek 39 Tabulka po přejmenování hrany „PROBABLY_SAME_OFFICER_AS“ na název „SAME_NAME_AS“ (Zdroj vlastní.)

Použití algoritmu pro zjištění blízkosti polohy ve středu není v rámci Panama Papers data setu příliš vhodné, a to z toho důvodu, že veškeré uzly a hrany netvoří jeden spojený celek, jako to je například v problematice sociálních sítí. V data setu se vykytují uzly, které pomocí hran tvoří úzké a uzavřené společnosti a značně tak narušují zjištění blízkosti polohy ve středu v rámci celého data setu.

Předvést algoritmus společných sousedů nelze, aby měl nějaký logický význam v rámci Panama Papers data setu.

Co však může být zajímavé je použití algoritmu náhodné procházky, kde výchozí bod neboli společnost je například *"MOSSACK FONSECA & CO. CZ, S.R.O"*, která má 1 550 sousedů. Dotaz je následující:

```
MATCH (i:Intermediary {name: "MOSSACK FONSECA & CO. CZ, S.R.O"})
CALL algo.randomWalk.stream(id(i), 2, 4, {path:true}) YIELD nodeIds
UNWIND nodeIds AS nodeId
RETURN algo.getNodeById(nodeId).name AS Name
```

Výsledná tabulka (obrázek 40) zobrazuje proces, kde algoritmus náhodné procházky zjišťuje čtyři různé cesty za pomocí dvou kroků.

Name
"MOSSACK FONSECA & CO. CZ, S.R.O"
"RAVOT LTD."
"MOSSACK FONSECA & CO. CZ, S.R.O"
"MOSSACK FONSECA & CO. CZ, S.R.O"
"EXCAVATORS PARTS LTD."
"MOSSACK FONSECA & CO. CZ, S.R.O"
"MOSSACK FONSECA & CO. CZ, S.R.O"
"MAXCOOPER AGENCY LTD."
"Ms. Christina Cornelia van den Berg"
"MOSSACK FONSECA & CO. CZ, S.R.O"
"KM UNITRADE CORP."
"MOSSACK FONSECA & CO. CZ, S.R.O"

Obrázek 40 Tabulka znázorňuje výsledky použití náhodné procházky, kde výchozí bod je společnost "MOSSACK FONSECA & CO. CZ, S.R.O" (Zdroj vlastní.)

Panama Papers data set obsahuje velké množství informací v podobě uzlů a hran, jenž mohou být analyzovaná a následně interpretována.

6.5 Shrnutí a vyhodnocení provedené analýzy

Analýza dat v rámci Neo4j probíhala na Panama Papers data setu získaného z Neo4j Sandboxu za pomocí Neo4j Serveru a dotazovacího jazyka Cypher ve vizualizačním prostředí Neo4j Browseru. I přesto, že Neo4j Sandbox nezpřístupňuje veškerá Panama Papers data, která byla reálně kdy získána, data set o celkové velikosti 942 MiB obsahuje dostatečné množství uzlů a hran pro provedení detailnější analýzy. Data set nabízí zajímavou provázanost dat, je tedy velmi snadné data určitým způsobem uchopit a dále s nimi pracovat.

Pomocí krátkých dotazů v dotazovacím jazyce Cypher byly získány základní informace pro seznámení s grafovou databází, od kterých se dále odvíjela celá analýza. Dále se analýza soustředila na konkrétní figurování určitých uzlů a hran v problematice Panama Papers. V rámci dataminingu nad grafovou databází byl použit i plugin grafových algoritmů, které usnadňovaly získávání některých informací.

Dotazy byly prováděny v řádech milisekund, pokud se však jednalo o zmíněné grafové algoritmy, jiné složitější dotazy či vykreslování grafů, jednalo se mnohdy i o řád sekund.

Výsledky dotazů bylo možné takřka ihned analyzovat a určitým způsobem interpretovat a to pomocí příjemného Neo4j Browser vizualizačního prostředí. Neo4j Browser totiž umožňuje přímo vkládat dotazy v dotazovacím jazyce Cypher, který svou syntaxí připomíná dotazovací jazyk SQL, a ihned je zpracovávat, jak v tabulkovém provedení tak i grafickém.

Společnost Neo4j umožňuje uživatelům pracující s Neo4j Sandboxu Panama Papers grafovou databázi rozšiřovat o důvěryhodné uzly a hrany či se zajímá například i o poukázání na zajímavé konexe mezi uzly. Tudíž pokud jsou nalezeny přínosné informace, data set může být rozšířen o více dat pro všechny uživatele, a to i přesto, že již proběhlo několik investigativních šetření této problematiky. Ovšem analýza Panama Papers má i individuální přínos, a to jak z pohledu zájmu o problematiku Panama Papers, tak i z pohledu dotazování a práce nad neustále se vyvíjející rozhraní Neo4j.

7 Závěr

Bakalářská práce podává teoretický přehled o NoSQL databázích se soustředěním na grafické databáze, kde byla vybrána Neo4j databáze pro další části práce, a dataminingu obecně. Práce dále popisuje, jakým stylem probíhá instalace vybrané Neo4j databáze, konkrétně Neo4j Community Serveru. Zahrnuje taktéž vizualizační nástroj Neo4j Browser pro okamžité vykreslení získaných a představuje ukázky použití dotazovacího jazyka Cypher nad databází, pro představení, jakým způsobem zmíněný dotazovací jazyk umožňuje získávání informací.

V rámci dataminingu a Neo4j databáze jsou vytvořeny ukázky použití grafových algoritmů, jako metody dataminingu, za pomocí instalovaného pluginu pro Neo4j nad daty získaných z volně přístupných zdrojů na Internetu, např. použití algoritmů pro získání nejkratší cesty mezi uzly grafu či algoritmus společných sousedů určitých uzlů.

Hlavní částí práce je provedení analýzy dat v rámci Neo4j. Byla vybrána data týkající se Panama Papers. Data byla získána pomocí Neo4j Sandboxu a představují záznamy společností a osob figurujících v Panama Papers. Nad zmíněnými daty pomocí dotazovacího jazyka Cypher byly provedeny jednoduché dotazy spolu s metodami dataminingu pro získání zajímavých informací, které mohou být využity pro další možné budoucí investigace či jako zajímavé postřehy, jež samotná společnost Neo4j oceňuje. Veškerá získaná data jsou vykreslena v grafech či zanesena v tabulkách a následně interpretovaná.

I. Summary and key words

This bachelor thesis deals with the area of NoSQL databases with a focus on graph databases and the application of data mining methods.

The main aim is to install a specific NoSQL graph database Neo4j together with the characterization of its key features and activities and to use data mining methods above the graph database. Methods of data mining are performed and analyzed the most typical examples of data obtained from the Internet. The used query language is Cypher.

The last part of this thesis is an analysis of selected data from Neo4j Sandbox with a focus on the Panama Papers in the Neo4j database. To present this example, tools as Neo4j Community Server and Cypher language are used. The results of this analysis are then evaluated for their possible further use.

Key words: NoSQL, graph database, data mining, Neo4j, Cypher

II. Seznam použitých zdrojů

2.1. System requirements - Chapter 2. Installation. (n.d.). Retrieved March 2, 2019, from https://neo4j.com/docs/operations-manual/3.5/installation/requirements/

4.4. The Closeness Centrality algorithm - Chapter 4. Centrality algorithms. (n.d.). Retrieved March 27, 2019, from https://neo4j.com/docs/graph-algorithms/current/algorithms/closeness-centrality/

6.7. The Random Walk algorithm - Chapter 6. Path finding algorithms. (n.d.). Retrieved March 28, 2019, from https://neo4j.com/docs/graph-algorithms/current/algorithms/random-walk/

8.2. The Common Neighbors algorithm - Chapter 8. Link Prediction algorithms. (n.d.). Retrieved March 27, 2019, from https://neo4j.com/docs/graph-algorithms/current/algorithms/linkprediction-common-neighbors/

Bartha, T. (2015). Grafové databáze a jejich aplikace na sociální sítě (Bakalářská práce). Jihočeská univerzita v Českých Budějovicích, České Budějovice.

Chapter 1. Introduction - The Neo4j Graph Algorithms User Guide v3.5. (n.d.). Retrieved March 26, 2019, from https://neo4j.com/docs/graph-algorithms/current/introduction/#_installation

Data Mining Concepts. (n.d.). Retrieved March 26, 2019, from https://docs.oracle.com/cd/B28359_01/datamine.111/b28129/process.htm#CHDFGCIJ

Efficient Graph Algorithms for Neo4j. Contribute to neo4j-contrib/neo4j-graph-algorithms development by creating an account on GitHub [Java]. (2019). Retrieved March 26, 2019 from https://github.com/neo4j-contrib/neo4j-graph-algorithms (Original work published 2017)

Graph Database Use Cases and Solutions. (n.d.). Retrieved March 4, 2019, from Neo4j Graph Database Platform website: https://neo4j.com/use-cases/

Hills, T. (2016). NoSQL and SQL Data Modeling: Bringing Together Data, Semantics, and Software (1st ed.). Basking Ridge, NJ: Technics Publications.

Hodler, A., & Needham, M. (2019). 4. Pathfinding and Graph Search Algorithms - Graph Algorithms [Book]. Retrieved March 27, 2019, from https://www.oreilly.com/library/view/graph-algorithms/9781492047674/ch04.html

Holubová, I., Kosek, J., Minařík, K., & Novák, D. (2015). Big Data a NoSQL databáze (1st ed.). Praha: Grada Publishing.

Hřivna, J. (2016). Grafové databáze - představení a ukázka užití (Bakalářská práce). Vysoká škole ekonomická v Praze, Praha.

Jalili Mahdi, Orouskhani Yasin, Asgari Milad, Alipourfard Nazanin, & Perc Matjaž. (2017). Link prediction in multiplex online social networks. Royal Society Open Science, 4(2), 160863. https://doi.org/10.1098/rsos.160863

Kenton, W. (2018). Panama Papers. Retrieved March 19, 2019, from Investopedia website: https://www.investopedia.com/terms/p/panama-papers.asp

Lažanský, J. (2014). Architektury databázových systémů. Retrieved from http://labe.felk.cvut.cz/vyuka/A3B33OSD/Tema-13-ArchitekturyDistribDBMS-OSD-4.pdf

Listinné akcie na doručitele / NFPK. (n.d.). Retrieved March 24, 2019, from http://www.nfpk.cz/listinne-akcie-na-dorucitele

Neo4j Graph Platform – The Leader in Graph Databases. (n.d.). Retrieved March 2, 2019, from Neo4j Graph Database Platform website: https://neo4j.com/

Neo4j Licensing Overview. (n.d.). Retrieved March 4, 2019, from Neo4j Graph Database Platform website: https://neo4j.com/licensing/

Obermaier, F., Obermayer, B., Wormer, V., & Jaschensky, W. (n.d.). All you need to know about the Panama Papers. Retrieved March 19, 2019, from Süddeutsche.de website: https://panamapapers.sueddeutsche.de/articles/56febff0a1bb8d3c3495adf4/

Panyko, T. (2013). NoSQL databáze (Bakalářská práce). Jihočeská univerzita v Českých Budějovicích, České Budějovice.

Řáda, J. (2017). Aplikace NoSQL databází v oblasti podnikových informačních technologií (Bakalářská práce). Unicorn College, Praha.

Rajkumar, P. (2014, August 20). 14 useful applications of data mining. Retrieved March 26, 2019, from Big Data Made Simple website: https://bigdata-madesimple.com/14-useful-applications-of-data-mining/

Real Datasets for Spatial Databases: Road Networks and Category Points. (n.d.). Retrieved March 28, 2019, from https://www.cs.utah.edu/~lifeifei/SpatialDataset.htm

Reprezentace grafů. (n.d.). Retrieved February 28, 2019, from https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=9312

Robinson, I., Webber, J., & Eifrem, E. (2013). Graph Databases (1st ed.). Sebastopol, CA: O'Reilly Media.

Šeleng, M., Laclavík, M., Dlugolinský, Š., & Hluchý, L. (2011). Dostupné škálovateľné riešenia pre spracovanie veľkého objemu dát a dátové sklady. Retrieved from http://laclavik.sk/publications/datakon_final_2011.pdf

Škrášek, J. (2015). Social Network Recommendation using Graph Databases (Diplomová práce). Masarykova univerzita, Brno.

Network datasets: Social circles. (n.d.). Retrieved March 28, 2019, from https://snap.stanford.edu/data/egonets-Twitter.html

Social network analysis: Centrality measures. (2014, December 3). Retrieved March 27, 2019, from Cambridge Intelligence website: https://cambridge-intelligence.com/keylines-faqs-social-network-analysis/

Strozzi, C. (1998). NoSQL: A Relational Database Management System. Retrieved March 2, 2019, from http://www.strozzi.it/cgi-bin/CSA/tw7/I/en_US/NoSQL/Home%20Page

Tiwari, S. (2011). Professional NoSQL (1st ed.). John Wiley & Sons.

Vardanyan, M. (2017, 6). Picking the Right NoSQL Database Tool. Retrieved February 28, 2019, from https://www.monitis.com/blog/picking-the-right-nosql-database-tool/

Vukotic, A., & Watt, N. (2015). Neo4j in Action (1st ed.). Shelter Island, NY: Manning Publications Co.

Walmart - Neo4j Graph Database Platform. (n.d.). Retrieved March 4, 2019, from Neo4j Graph Database Platform website: https://neo4j.com/case-studies/walmart/

Walmart and eBay adopt graph database | News | Retail Technology. (n.d.). Retrieved March 4, 2019, from http://www.retailtechnology.co.uk/news/5187/walmart-and-ebay-adopt-graph-database/

Yuhanna, N. (2017). Vendor Landscape: Graph Databases.

III. Seznam obrázků

IV. Seznam tabulek