VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
DEPARTMENT OF INTELLIGENT SYSTEMS

GENEROVÁNÍ RODOKMENŮ Z MATRIČNÍCH ZÁZNAMŮ

FAMILY TREES MAKING FROM PARISH RECORDS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. LUCIA TUŠIMOVÁ**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. JAROSLAV ROZMAN, Ph.D.**
SUPERVISOR

BRNO 2020

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav inteligentních systémů (UITS) Akademický rok 2019/2020

Zadání diplomové práce

Studentka: **Tušimová Lucia, Bc.**
Program: Informační technologie Obor: Inteligentní systémy
Název: **Generování rodokmenů z matričních záznamů**
Family Trees Making from Parish Records
Kategorie: Umělá inteligence

Zadání:

1. Nastudujte obor genealogie a materiály v ní používané (matriční knihy, urbáře, katastry, atd). Nastudujte práce zabývající se automatickým propojováním záznamů do větších rodokmenů.
2. Na základě nastudované literatury navrhněte způsob, jak záznamy propojovat do větších celků. Počítejte s tím, že pro danou oblast nebudou v danou chvíli k dispozici všechny záznamy - ty budou teprve průběžně přibývat. Dále systém navrhněte jako pravděpodobnostní, tzn. dítě může mít více rodičů, pro které se budou počítat pravděpodobnosti, které se budou s postupně přidávanými záznamy měnit. Data se budou načítat z relační databáze a ukládat do grafové databáze.
3. Navržený systém implementujte.
4. Testování proveďte na testovací sadě dodané vedoucím.

Literatura:

- Dintelman, S., Maness, T.: Reconstituting the Population of a Small European Town Using Probabilistic Record Linking: A Case Study, Family History Technology Workshop, BYU 2009
- Malmi, E., Rasa, M., Gionis, A.: AncestryAI: A Tool for Exploring Computationally Inferred Family Trees, Proceedings of International World Wide Web Conference Comittee, 2017

Při obhajobě semestrální části projektu je požadováno:

- První dva body zadání.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Rozman Jaroslav, Ing., Ph.D.**
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 3. června 2020
Datum schválení: 31. října 2019

Abstrakt

Táto práca rozoberá obor genealógie, rôzne druhy záznamov a údaje v nich. V práci je opísaná tematika porovnávania a klasifikovania záznamov. Ďalej rozoberá návrh a implementáciu výsledného systému. Vyvinutý systém prepája osoby z matričných záznamoch do väčších rodokmeňov. Tie sú následne uložené vo forme grafovej databázy. Úspešnosť prepájania záznamov bola testovaná nad poskytnutými dátovými sadami.

Abstract

This work discusses the field of genealogy, different types of records and data in them. The thesis describes the topic of comparison of data and record linkage. It further it also discusses the design and implementation of the resulting system. The developed system connects people from parish records to larger pedigrees. These are then stored in the form of a graph database. The success of the interconnection of records was tested on the provided data sets.

Kľúčové slová

genealógia, matričné záznamy, prepájanie záznamov, grafové databázy

Keywords

genealogy, parish records, record linkage, graph databases

Citácia

TUŠIMOVÁ, Lucia. *Generování rodokmenů z matričních záznamů*. Brno, 2020. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jaroslav Rozman, Ph.D.

Generování rodokmenů z matričních záznamů

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracovala samostatne pod vedením pána Ing. Jaroslava Rozmana Ph.D.. Uviedla som všetky literárne pramene a publikácie, z ktorých som čerpala.

.....
Lucia Tušimová
10. júna 2020

Poďakovanie

V úvode by som rada poďakovala, vedúcemu mojej práce, Ing. Jaroslavovi Rozmanovi Ph.D., za jeho pomoc a rady pri vedení diplomovej práce.

Obsah

Kapitola 1

Úvod

Všetci sme sa určite už zamýšľali nad tým, čím boli naši predkovia. Možností je veľa, mohli to byť udatní rytieri, prostí roľníci, alebo významní šľachtici. Aby sme sa však dozvedeli pravdu o našom pôvode, musíme začať tvorbou rodokmeňu. Tak ako nám technika a internet zjednodušujú mnoho aspektov života ani toto odvetvie nezaostáva. Matriky sa v Českej republike začali digitalizovať v roku 2007[2]. Postupne prechádzajú všetky matričné záznamy do digitálnej a teda podstatne dostupnejšej formy. Vďaka tomuto pokroku sa historické záznamy uchovávajú aj pre ďalšie generácie. Ukončenie digitalizácie matrík pre celú Českú republiku bolo pôvodne plánované v roku 2011. Vzhľadom na to, že sa jedná o obrovské množstvo historických záznamov, digitalizácia prebieha dodnes.

Prístup k matričným záznamom z pohodlia domova je obrovským krokom vpred. Tvorba rodokmeňov sa tak stáva dostupnou pre každého. Keď však už máme také množstvo dát v elektronickej podobe, mohli by sme si naše pátranie opäť o čosi zjednodušiť. Ich spracovanie a následne automatické prepojenie do rodokmeňov, by nám mohlo doslova ušetriť roky strávené hľadaním informácií po matričných úradoch.

Moja práca sa zaoberá práve spomínaným prepojovaním matričných záznamov do rodokmeňov. Cieľom je vytvoriť program, ktorý bude schopný prepájať osoby v matričných záznamoch. Súčasťou rodokmeňu sú všetky dostupné informácie od pohlavia a dátumu narodenia až cez bydlisko či povolanie. Čím viac záznamov bude systém obsahovať, tým väčšie rodokmene bude schopný vytvoriť.

Prvá časť diplomovej práce sa zaoberám teoretickými poznatkami. Definujeme základné pojmy z genealógie a podrobne si rozoberieme matričné zdroje. Uvedieme si príklady možných zdrojov genealogických dát. Následne si špecifikujeme funkcionalitu programu. Postupne si priblížime jednotlivé časti, z ktorých sa program skladá.

V ďalšej časti sa venujeme podrobnému návrhu programu. Na základe existujúcej relačnej databázy si vytvoríme vlastnú grafovú databázu. Detailne si prejdeme a definujeme všetky informácie, ktoré môžeme získať. Následne analyzujeme jednotlivé dáta a špecifikujeme si porovnávacie metódy, ktoré budeme využívať pri ich spracovaní. Ako posledné si zvolíme vhodný spôsob na zaznamenávanie výsledkov.

Na záver práce sa venujeme samotnej implementácii a jej testovaniu. Oboznámime sa s použitými technológiami a detailne si prejdeme algoritmom. Celý systém napokon otestujeme s použitím testovacích dát a zhodnotíme jeho úspešnosť pri vytváraní rodokmeňov.

Kapitola 2

Štúdium teórie

V nasledujúcich kapitolách sú popísané teoretické informácie, ktoré som potrebovala naštudovať pre túto prácu. V prvej časti sa venujem pojmom genealógia či rodokmeň. Ďalej rozoberiem matričné záznamy a ďalšie pramene používané v genealógii. V poslednej časti tejto sekcie sa budem venovať spôsobom slúžiacim na prepojovanie záznamov do väčších rodokmeňov. Všetky uvedené údaje sa budú týkať Českej republiky.

2.1 Genealógia

Slovo genealógia pochádza z rímskych slov *génos* = rod a *logos* = veda. História genealógie nesiaha však iba do starovekého Ríma. Prvé genealogické prvky môžme pozorovať už v starovekom Egypte a to tabuľky s poradím panovníkov, alebo záznamy z Biblie zo starého zákona. Nájdeme ju aj v starovekom grécku, kde si významný občania odvodzovali svoj pôvod od bohov. Dokonca aj v stredovekej Európe si šľachta s obľubou dávala vypracovávať rodokmene. Genealógia je teda pevne prepojená s našimi dejinami.

Zjednodušene môžeme povedať, že genealógia je vedou o rode a rodine. Nezaoberá sa len štúdovaním rodinnej histórie, ale aj vytváraním rodokmeňov a hľadaním pôvodu. Je zaradená medzi pomocné historické vedy. Skúma vzťahy medzi jedincami, ktoré vyplývajú z ich spoločenského rodového pôvodu. Informácie o genealógii v tejto kapitole ako aj v nasledujúcich podkapitolách sú prebrané z týchto zdrojov [6, 12, 16].

2.1.1 Základné pojmy

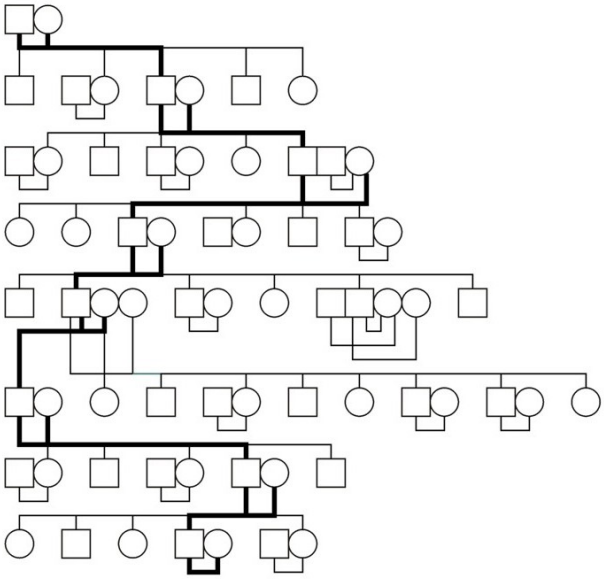
Medzi základné štruktúry používané v genealógii, ktoré si rozoberieme patrí rodokmeň, vývod a rozrod.

Rodokmeň

Ak máte záujem pátrať v rodinnej histórii pravdepodobne začnete vytvorením rodokmeňu. Zatiaľ, čo v histórii boli rodokmene záležitosťou prevažne šľachtických rodov, v súčastnosti sú rozšírené medzi širokú verejnosť. Ako prvý krok sa zväčša pri vytváraní rodokmeňu vychádza z rozhovorov a pamätí žijúcich ľudí. Kam už pamäť nesiaha, máme ďalšie možné zdroje a to rôzne historické dokumenty, ako napríklad matričné knihy, urbáre, katastre. Tie si podobnejšie rozoberieme neskôr 2.1.2.

Pod pojmom rodokmeň, alebo rozvoj sa rozumie genealogická tabuľka. Tá uvádza predkov určitej osoby zväčša len po otcovskej línii, teda nositeľov rovnakého priezviska. Údaje

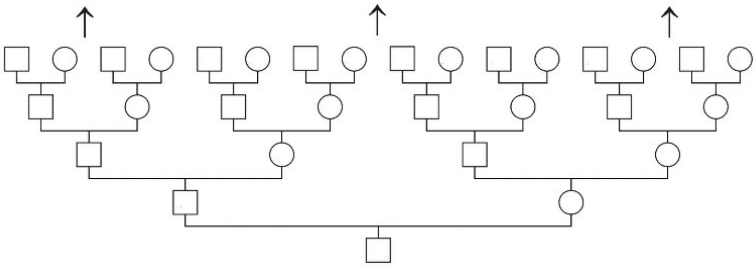
o manželkách z pravidla nebývajú veľmi obšírne. Najstarší pár, ktorý sa nám podarí nájsť, sa potom považuje za zakladateľov rodu.



Obr. 2.1: Schéma rodokmeňu [19]

Vývod

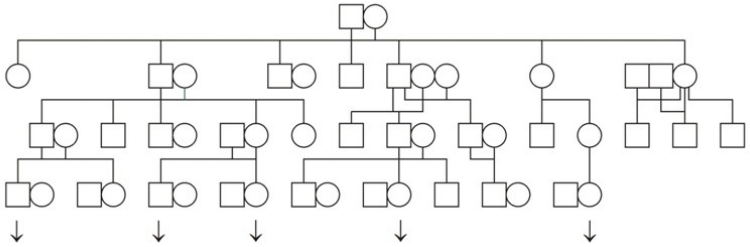
Vývod sa zaoberá všetkými predkami mužskej aj ženskej línie, od konkrétnej osoby. Zapisujú sa iba pokrvní príbuzní a teda iba páry, ktoré boli zodpovedné za splodenie danej osoby. Manželstvo teda nie je v tomto prípade podmienkou. Smeruje od prítomnosti, až tak ďaleko do minulosti do akej sa náš genealogický výskum dostane. Ten je však úzko spätý s kvalitou a dostupnosťou zachovalých materiálov. Vývod býva zakreslený v podobe stromu 2.2, kde koreň zobrazuje danú osobu, ktorej sa výskum týka. V prípade ak budeme brať v úvahu iba otcovskú líniu, tak ide o agnátny vývod. Naopak ak zanedbáme otcovskú a bude nás zaujímať iba materská línia pôjde o kognátny vývod.



Obr. 2.2: Schéma vývodu [19]

Rozrod

Pri rozrode sa jedná o zobrazenie všetkých potomkov najstaršieho dohľadaného konkrétneho páru. Súpis obsahuje všetkých potomkov nesúcich rovnaké priezvisko teda aj nemanželských detí. Čo však môže znamenať, že sa bude jednať o tisíce záznamov. Pri niektorých rodoch však môžeme pozorovať opačný jav a to postupné vymieranie. Vo všeobecnosti však ide o najzložitejšiu štruktúru z vyššie zmienených.



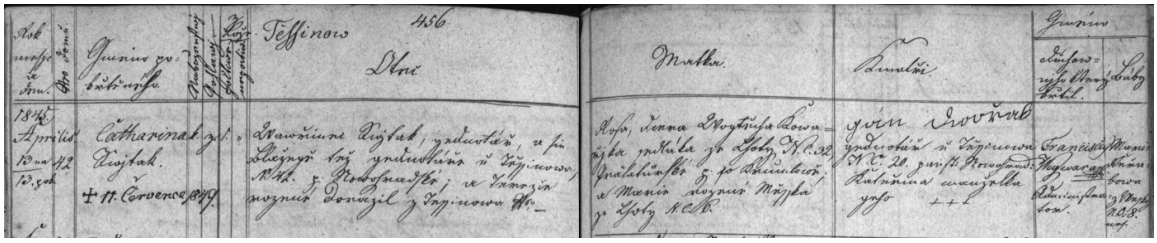
Obr. 2.3: Schéma rozrodu [19]

2.1.2 Matričné záznamy

Matriky sú základným zdrojom informácii pri bádaní v genealógii. Prvé boli vedené cirkevnými inštitúciami. Avšak od roku 1950 boli premiestnené pod štátnu správu. od tejto doby sú vedené iba štátne matriky. Do týchto kníh sa zaznamenávali všetky dôležité udalosti v živote človeka. Nachádza sa tam záznam o narodení, krste, birmovke, svadbe, partnerstve a smrti. Najstaršia zachovaná matrika na území Českej republiky vznikla v polovici 16. storočia a jedná sa o evanjelickú matriku. I keď nariadenie o vedení matrík vzniklo už v roku 1563 presadzovalo sa iba veľmi pomaly. Rozsah prvých matrík môžme prirovnať iba k zoznamu veriacich, ktorí patrili pod určitého duchovného. Matriky sa delia podľa veku na takzvané živé a mŕtve. Živé matriky obsahujú záznamy mladšie ako 100 rokov, ak sa jedná o narodenie. V prípade svadby alebo úmrtia je to 75 rokov. Záznamy v mŕtvych matrikách by mali byť teda len o mŕtvych ľudoch a je k nim umožnený prístup.

Matrika pokrstených/narodených

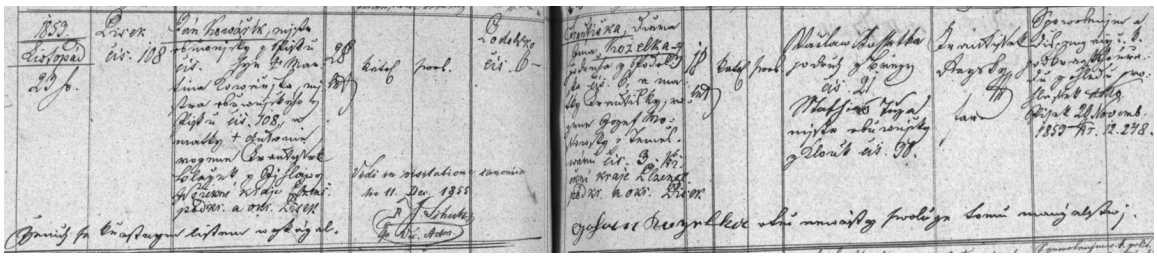
Základné údaje, ktoré boli uvedené v matrike o narodení dieťaťa bolo jeho meno, dátum narodenia, pohlavie, meno otca, matky a kmotra. V minulosti kňaz často zaznamenával deň krstu a nie deň narodenia. Z dôvodu vysokej úmrtností malých detí sa však krst častokrát konal čo najskôr po pôrode. V niektorých prípadoch sú však uvedené obidva a sú zapísané nad sebou. Pri mene matky a dieťaťa sa do 18. storočia jednalo iba o krstné mená a priezviská boli odvodzované od mena otca. Postupom času sa tieto záznamy rozrastali o podrobnejšie údaje o dieťati, napríklad či išlo o nemanželské dieťa, miesto narodenia alebo štátnu príslušnosť. Rovnako sa podrobnejšie zaznamenávali aj údaje o rodičoch a to bydlisko, náboženstvo a povolanie. Pridali sa záznamy o starých rodičoch, krstiteľovi, pôrodnej babe a svedkovi. Väčšina týchto údajov sa pridala v 19. storočí. Zaujímavosťou je, že v prípade, ak dieťa umrelo pri pôrode, alebo veľmi malé nasledujúce dieťa dostávalo rovnaké meno. Samozrejme bral sa ohľad na pohlavie dieťaťa prípadne poverčivosť, ak zomrelo viac detí s rovnakým menom. Prvorodené deti dostávali často mená po svojich otcoch alebo matkách.



Obr. 2.4: Římskokatolický farní úřad Jílovice, matrika narozených 1804-1865 [11]

Matrika oddaných

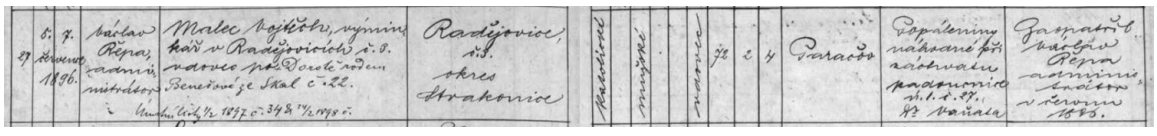
Základom matričného záznamu, ktorý mapoval svadbu bolo meno ženícha, nevesty, svedkov a kňaza. V minulosti sa stávalo, že žene bolo zapisované len krstné meno a priezvisko prebrala po svojom mužovi. Podobne, ako keď sa jednalo o zápis narodených, tak aj v tomto prípade sa časom zápisy stávali podrobnejšie. Pribudli rodičia ženícha a nevesty, ale aj podobnejšie údaje o všetkých zúčastnených, teda bydlisko, dátumy narodenia, povolania a náboženstvo. V minulosti bola rola otca nevesty veľmi dôležitá, keďže ak nedovŕšila dospelosť, musela mať súhlas na sobáš. Ženy často pristupovali k sobášu ešte pred dovŕšením dospelosti. V minulosti bol však vek pre vstup do dospelosti vyšší a to 21 až 24 rokov. Svedkovia zohrávali aj stále zohrávajú neoddeliteľnú súčasť svadby. V minulosti bolo hlavnou úlohou svedkov zabezpečiť, aby sa o svadbe dozvedeli rodičia a vylúčila sa možnosť utajeného zväzku.



Obr. 2.5: Římskokatolický farní úřad Chřešťovice, matrika oddaných 1818-1864 [11]

Matrika zosnulých

Pri zápisoch úmrtia sa jednalo o strohejšie zápisy, ktoré poskytovali len základné informácie. V prípade ak zosnulého identifikovali, zapisovalo sa jeho meno, bydlisko a dátum narodenia. Väčšina informácii sa však týkala smrti samotnej a to jej príčina, dátum či miesto.



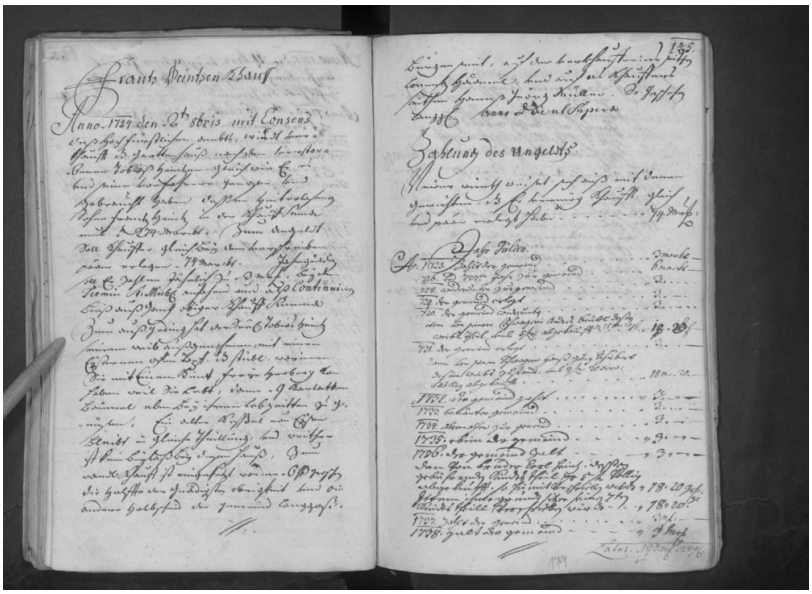
Obr. 2.6: Římskokatolický farní úřad Paračov, matrika zemřelých 1861-1929 [11]

2.1.3 Ďalšie pramene

Okrem matrík sú v archívoch uložené aj mnohé ďalšie pramene, ktoré nám môžu pomôcť pri rozširovaní údajov. Išlo o rôzne písomnosti, týkajúce sa vojakov či už zápisy, alebo kmeňové listy. Okrem toho farské archívy, v ktorých sú zaznamenané zoznamy duší, školské písomnosti, cechovné archívy, podnikové písomnosti či mnohé ďalšie. Podrobnejšie si rozoberieme tie významnejšie zdroje a to gruntovné knihy, urbáre, archívy miest a obcí, kataster a sčítanie ľudu.

Gruntovné knihy

V súčastnosti známe skôr ako pozemkové knihy. Slovo gruntovné pochádza z ich pôvodne nemeckého názvu Grundbuch. Prvé takéto knihy na území Českej republiky sa datujú už od 12. storočia. Takéto knihy sú charakteristické pre naše územie, pretože v iných krajinách strednej Európy sa používali prevažne listy vlastníctva. Zapisovali sa do nich teda údaje o prevode a predaji pozemkov a nehnuteľností. V minulosti vlastnila pôdu iba šľachta a obyčajní ľudia si ju od nej mohli len prenajímať a obhospodarovať. Keď v roku 1848 zrušili poddanstvo, tak sa tieto knihy zmenili na spomínané pozemkové a ľudia už mohli pôdu skutočne vlastniť. Väčšinou usadlosť dedili synovia po otcoch. Avšak v niektorých prípadoch mohol odkúpiť usadlosť od vdovy jej nový manžel. Prípadne mohli vymeniť hospodára, ak sa niekomu nedarilo[4].



Obr. 2.7: Príklad z Gruntovej knihy

Urbáre

Ide o súpisy dávok peňazí a naturálií, ktorú vyberala vrchnosť od poddaných. Prevažne dvakrát do roka a to na jar a na jeseň. Urbáre sa nachádzali u nás skôr ako pozemkové knihy. Sú vedené podľa jednotlivých panstiev. Najstaršie pochádzajú už z roku 1378. Zachovalosť urbárov závisela od daného panstva a nie od celého štátu, preto je ich stav veľmi subjektívny. Ak sa v urbároch zameriame na genealogické dáta, z ktorých by sme mohli

prepájať záznamy, nájdeme tam meno a priezvisko hospodára. Nie sú tam zmienky o iných obyvateľoch hospodárstva ani podrobnejšie informácie o hospodárovi.

Kataster

Tieto pramene vznikali z daňových dôvodov. Najstarší dochovaný kataster je takzvaná berná rula, ktorá pochádza z roku 1654. Tento názov pochádza z staročeského slova berně, čo znamená daň. Boli v nej zapísané súpisy hospodárov, ich pozemky a hospodárske zvieratá. Najstarším moravským súpisom hospodárov je Lánový rejstrík, ktorý pochádza z roku 1656. V priebehu rokov sa tieto katastre vylepšovali, stávali sa presnejšími a lepšími na daňové účely. Po bernej rule a lánových rejstríkoch nasledoval Tereziánsky kataster, ktorý tu bol za vlády Márie Terézie. Z čias jej syna Jozefa II. prešiel opätovne kataster veľkou reformou a nazýval sa Jozefínsky. Stabilný kataster bol vydaný v roku 1817, ktorý zjednodušil a spresnil vymeranie dane pre všetky pozemky. Významnejšie oblasti napríklad mestá boli zmapované podrobnejšie.

Sčítanie ľudu

Už od stredoveku chcela mať šľachta prehľad o svojich obyvateľoch. Avšak prvý súpis obyvateľstva celej krajiny prebehol v roku 1754. Pochopiteľne je pomerne dosť nepresný. Za moderné sčítania ľudu môžme považovať tie od roku 1857. Môžu nám ponúknuť množstvo informácií o našich predkoch a to povolanie, národnosť, vierovyznanie, gramotnosť, hospodárske vybavenie, ale aj počet potomkov a mnohé iné. Práve tieto informácie sú také cenné, pretože sa nenachádzajú v žiadnych iných historických materiáloch. Sčítanie prebiehalo takmer vždy v 10 ročných intervaloch.

2.1.4 Chyby v záznamoch

Ako už bolo spomenuté údaje, ktoré sa zapisovali do matrík sa časom menili. V rôznych oblastiach zmeny prichádzali postupne a v minulosti to mohlo trvať aj niekoľko rokov. Rôzny zapisovatelia viedli rôzne podrobné záznamy a preto, kým sa matričné záznamy stali jednotnými, prešli dlhú cestu.

Okrem rozmanitých údajov v matrikách v nich často nachádzame aj mnohé nepresnosti. Tieto vychádzali napríklad z rôznych národností žijúcich na území Česka. Záznamy sa v minulosti zapisovali češtinou, ktorú používali obyčajní ľudia. Na zapisovanie sa používala aj latinčina ako jazyk duchovenstva a tiež nemčina, ktorú používali úradníci. Niektoré zápisy boli dvojjazyčné a v niektorých prípadoch rôzne časti dokonca v troch jazykoch. Stretneme sa teda aj s prekladmi mien a priezvisk do daného jazyka.

Keďže sa stará čeština gramaticky líši od tej súčasnej, nachádza sa v matrikách množstvo chýb. Či už ide o zámenu y a i, chýbajúcu diakritiku, nahrádzanie písmen v a j inými, alebo rôzne iné prvky jazyka, ktoré boli v minulosti bežne používané. Chyby však mohli vznikať aj z oveľa banálnejších dôvodov a to je zlá čitateľnosť, alebo rôzne varianty zápisu slov. Všetky tieto prvky sťažujú čitateľnosť záznamu a teda aj bádanie v minulosti rodov.

Príklady niektorých zápisov:

- Jiří - Jirzi, Girzj, Gyrzy
- Dvořák - Dworzak
- Kovář - Schmidt

- Mišun - Mischun
- Ouředník - Auředník
- Václav - Waczlawy
- ženích - zienich

2.1.5 GEDCOM

Internet ponúka množstvo nástrojov na tvorenie rodokmeňov. Formát GEDCOM je to, čo tieto nástroje spája. Jedná sa o skratku z Genealogical Data Communicaton, ide o súborový formát zapisovania genealogických dát. Bol vyvinutý Cirkvou Ježiša Krista Svätých posledných dní inak známych ako mormóni. I keď ide o už dlhodobo neaktualizovaný formát, posledná aktualizácia vyšla 28.12.2001 [1], je stále najpoužívanejší. Tieto súbory sú čisto textové, často vo formáte ASCII. Obsahujú genealogické informácie o osobách, ale aj záznamy, ktoré tieto osoby navzájom prepojujú [14].

Záznam GEDCOM sa skladá z hlavičky (header), sekcie záznamov (records) a špeciálneho koncového záznamu (trailer). V hlavičke nájdeme všeobecné informácie o súbore, ale aj údaje o programe v ktorom súbor vznikol. Môže teda obsahovať napríklad verziu štandardu, jazyk, dátum, alebo kódovanie. Po hlavičke už nasledujú jednotlivé záznamy o osobách (INDI record), o rodine (FAM record), alebo o zdrojoch informácií (SOUR record). Jednotlivý záznam o osobe môže okrem mena obsahovať množstvo podrobných informácií, ako miesto a dátum narodenia či úmrtia, alebo zamestnanie. Záznamy rodiny spájajú jednotlivé osoby do vzťahov. Posledný záznam je TRLR, ktorý označuje koniec súboru.

```
0 HEAD
1 GEDC
2 VERS 5.5.1
1 CHAR UTF-8
1 LANG Czech
1 SOUR MYHERITAGE
2 CORP MyHeritage.com
1 DEST MYHERITAGE
1 DATE 14 OCT 2019
1 FILE test
0 @I1@ INDI
1 NAME Petr /Novák/
1 SEX M
1 FAMS @F1@
1 BIRTH
2 DATE 1 JAN 1960
0 @I2@ INDI
1 NAME Karla /Svobodová/
1 SEX F
1 FAMS @F1@
1 BIRTH
2 DATE 1 JAN 1965
0 @I3@ INDI
1 NAME Tomáš /Novák/
1 SEX M
1 FAMC @F1@
1 BIRTH
2 DATE 1 JAN 1990
0 @F1@ FAM
1 HUSB @I1@
1 WIFE @I2@
1 MARR
1 CHIL @I3@
0 TRLR
```

Obr. 2.8: Príklad súboru GEDCOM

Na obrázku 2.8, môžme vidieť nenáročný príklad takéhoto súboru. V hlavičke môžme vidieť, že bol vytvorený pomocou stránky MyHeritage. Následne bol však zjednodušený z dôvodu prehľadnosti. Pre jednoduchosť sa v ňom nachádzajú iba 3 osoby s len 3 charakteristickými údajmi. Každá postava má priradený identifikátor. Ďalej pri nich nájdeme údaj

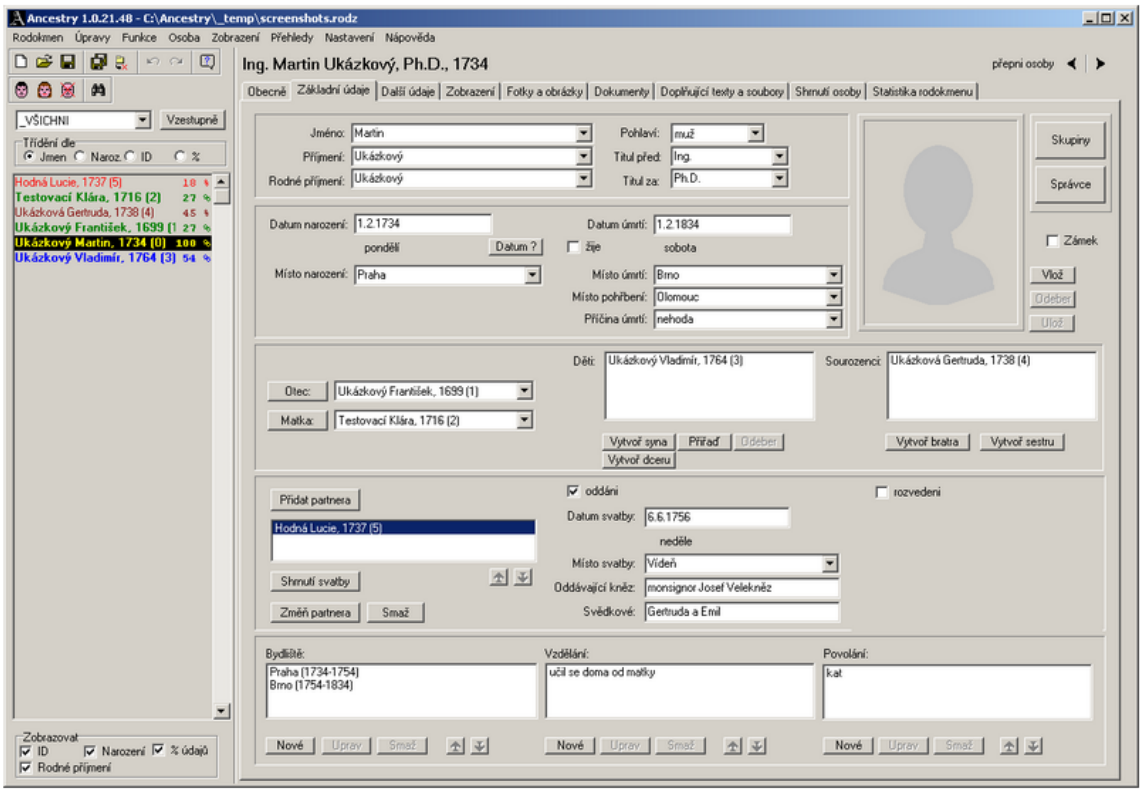
o dátume narodenia, pohlaví a priradení do rodiny. Pre zjednodušenie nie je pridaných viac charakteristických údajov. Tieto záznamy by však mohli byť výrazne podrobnejšie. V príklade sa nachádzajú manželia Petr a Karla, ktorí majú dieťa Tomáša. Tento vzťah medzi osobami je určený v zázname FAM. Pomocou značiek HUSB WIFE a MARR a ID osôb je vytvorené manželstvo. Následne mu je priradené jedno dieťa, ale ich počet je teoreticky neobmedzený. Na konci súboru sa musí samozrejme nachádzať záznam TRLR. Na začiatku každého záznamu je číslo, ktoré určuje stupeň zanorenia. Záznamy najvyššej úrovne sú značené číslom 0. Z obrázka 2.8 vidíme príklady HEAD, INDI, FAM no môžu to byť napríklad aj REPO alebo NOTE.

2.1.6 Dostupné riešenia

Na vytváranie rodokmeňov existuje množstvo programov. Rozoberieme si dva najznámejšie a to Ancestry a MyHeritage.

Ancestry

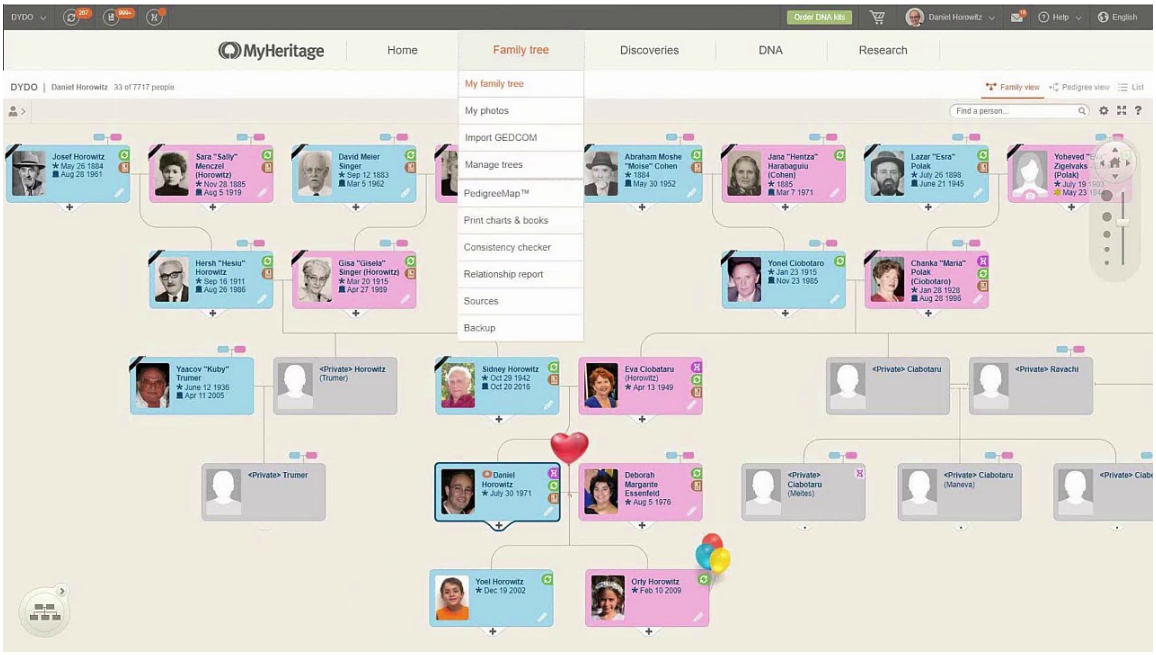
Ancestry [8] je český genealogický databázový program. Ponúka možnosť vytvorenia rodokmeňu, prepájania rodokmeňov, pridávanie fotografie i iné informácie k jednotlivým osobám. K ďalším možnostiam tohto programu patrí grafické zobrazenie rodokmeňu, vývodu či rozrodu. Vytvorený bol v roku 2003 a jeho vývoj i keď pomalšie pokračuje do súčastnosti. Jeho používanie je úplne zadarmo, čo je veľkou výhodou.



Obr. 2.9: Ukážka programu Ancestry

My Heritage

Pri MyHeritage ide o sociálnu sieť zameranú na rodinu. Je teda možné si vytvoriť svoj rodokmeň, zdielať fotografie, alebo vyhľadávať predkov. Základná verzia je bezplatná, má však výrazné obmedzenia. Po prejdení na Premium verziu sa sprístupnia nájdené zhody v systéme, v bezplatnej verzii sú sprístupnené iba útržky týchto informácii. Technológia Smart Matches porovnáva informácie z rodokmeňa s inými rodokmeňmi a dokáže ich prepájať [15]. Pomocou ďalšej technológie Record Matches spoločnosť zozbierala množstvo voľne prístupných materiálov, ktoré ďalej využíva pri rozširovaní rodokmeňov užívateľov. Ďalej využíva DNA testy na odhady etnického pôvodu, alebo zdravotné genetické riziká. Okrem toho ponúka aj genealogický software Family Tree Builder, ktorý umožňuje užívateľom pracovať bez pripojenia na internet. Ponúka vytváranie rodokmeňov, nahrávanie fotografií, prezeranie máp a štatistík. Tento program je veľmi jednoduchý a intuitívny na používanie.



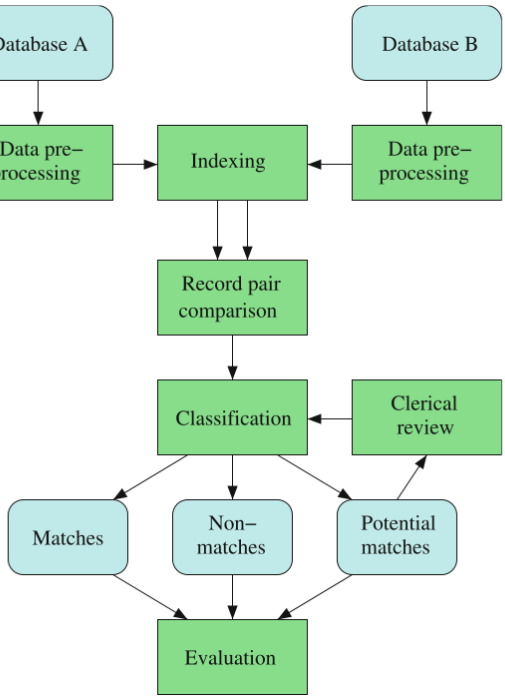
Obr. 2.10: Ukážka programu MyHeritage

2.2 Automatické prepojovanie do záznamov

Vo svete existuje mnoho menších výskumov, ktoré sa snažia rôznymi spôsobmi prepájať genealogické dáta. Pričom sa zameriavajú na jednotlivé miesta alebo rodinné línie. V článku [7] rekonštruujú populáciu malého európskeho mesta, za použitím pravdepodobnostnej analýzy2.2.5. Pričom ide o jeden z najbežnejších prístupov pri prepájaní záznamov. Podobné prístupy sa dajú následne použiť na veľké populácie. Ďalšie algoritmy na prepájanie záznamov sú vyvinuté na základe porovnania mien, priezvisk, dátumov a miest.

Následne si uvedieme, ako budeme s genealogickými dátami zaobchádzať. Prvým krokom ešte pred samotným prepojovaním je predspracovanie dát. Teda úprava dát, ktoré sme dostali do podoby s ktorou budeme ďalej pracovať. Ďalšou časťou je indexovanie, respektíve rozdelenie celej databázy do menších celkov. Zaoberáme sa tým z dôvodu zníženia kvadratickej náročnosti porovnávania. Až po vykonaní týchto krokov prichádza na rad samotné

porovnávanie. Keďže považujeme náš systém za pravdepodobnostný pri porovnávaní nám budú vznikať aj prípady nazývané potencionálne zhody. V prípade, ak je zhoda dostatočne vysoká, môžme prepojiť osoby. Posledným krokom je vyhodnotenie systému.



Obr. 2.11: Základný proces párovania dvoch databáz. [5]

2.2.1 Analýza dát

Prvým krok, ktorý musíme vykonať je rozobrať dáta na užitočné a nepotrebné. Intuitívne vieme, že určite pôjde o meno, priezvisko a pohlavie. Rovnako dátum a miesto narodenia budú ďalšími dôležitými ukazovateľmi. Ak nebudeme mať presný dátum narodenia, budeme sa snažiť odhadnúť interval. Napríklad narodenie matky z krstu jej dieťaťa. Ak nám neskôr príde presný údaj, použijeme ho. To nám zníži počet porovnávaní, keďže nebudeme musieť porovnávať osoby, ktoré boli v čase vzniku záznamu mŕtve. Z jedného údaju o krste môžme získať údaje až o 22 osobách. O samotnom dieťati, rodičoch, starých rodičoch, prarodičoch po starých mamách, krstných rodičoch a ich príbuzných, pôrodnej babe, krstiteľovi, a manželovi matky dieťaťa. Z historického hľadiska je dobrým ukazovateľom aj adresa osoby, keďže naši predkovia sa nesťahovali tak často. Podobne aj vierovyznanie, alebo povolanie. Nebolo zvykom, že by ho v rámci života človek často menil.

Údaje ktoré nás budú zaujímať o každej osobe:

- Meno
- Priezvisko
- Titul
- Pohlavie
- Národnosť
- Dátum narodenia
- Dátum krstu
- Interval dátumu narodenia

- Miesta kde osoba žila
- Vierovyznanie
- Dátum úmrtia
- Interval dátumu úmrtia
- Miesto úmrtia
- Povolanie
- Rodinný príslušníci

Hlavnou úlohou však bude prepájanie záznamov. Budeme mať dva základné prepojenia a to manželstvo a vzťah dieťa rodič. Tieto záznamy logicky vyplývajú aj zo záznamov v matrikách. Postupne sa budú menšie celky prepájať do väčších.

2.2.2 Predspracovanie dát

Okrem rozdielov, ktoré sme si spomenuli v kapitole 2.1.4, vznikajú chyby aj pri prevádzaní dát do elektronickej podoby. Na to využívame 2 metódy a to najmä metódu ručného prepisu, alebo metódou optického rozpoznávania znakov (OCR). I keď je ručné prepisovanie historických záznamov menej chybové je pomalšie. Po prevedení dát do elektronickej podoby je potrebné, pripraviť ich na ďalšiu prácu s nimi. Technika, ktorá sa využíva na prípravu, alebo čistenie dát sa nazýva Data cleansing. Je to proces odstránenia, alebo opravenia nepresných a chybných záznamov. Po vykonaní tohto postupu, by mali byť dáta konzistentné s ostatnými údajmi v systéme.

2.2.3 Zhlukovanie

V kapitole 2.1.4 sme si ukázali, že významovo rovnaké slovo môže byť zapísané rôznym spôsobom. Prípadne sa ľahko môže vyskytnúť iná chyba. Na odstránenie týchto chýb využijeme zhlukovanie. Na jeho základe nájdeme zhluky podobných mien napríklad Peter a Petr. Tieto mená následne upravíme na rovnaké. Táto metóda by nám mala pomôcť opraviť chyby v menách, priezviskách či názvoch miest.

2.2.4 Porovnávanie záznamov

Na to, aby sme vedeli spojiť dva záznamy, najskôr musíme vedieť, ako veľmi sa na seba podobajú. Samozrejme, ak nájdeme dva úplne rovnaké záznamy vieme, že ide o tú istú osobu. Ako sa však matričné záznamy s časom vyvíjali aj ich obsah sa menil. V starších záznamoch sa mnohokrát stáva, že časť záznamu chýba prípadne je chybne zapísaná. Preto aj záznamy, ktoré majú v niektorých atribútoch rôzne hodnoty môžu odkazovať na tú istú osobu. Našou úlohou bude teda zistiť, ako veľmi sa dané hodnoty podobajú.

Porovnávanie slov

Zhlukovú analýzu použijeme na úpravu niektorých typov dát. Pôjde o krstné mená, priezviská, povolania a názvy miest. Avšak nie pre všetky dáta je zhlukovanie vhodnou metódou. Ak ide o názvy ulíc, ktoré môžu byť rôznorodé, zhlukovaním by sme stratili chcené rozdiely. Na porovnávanie reťazcov znakov sa využívajú rôzne metódy ako napríklad Hammingova, Jarova, alebo Jaro-Winklerova vzdialenosť. My si predstavíme Levenshteinovu vzdialenosť, ktorá je založená na editačnej vzdialenosti.

Levenshteinovu vzdialenosť vymyslel ruský matematik Vladimit Levenshtein v roku 1965 [13, 18, 5]. Ako už bolo spomenuté, ide o meradlo vzdialenosti medzi dvoma slovami. Algoritmus dokáže nájsť najmenšie množstvo jedno znakových operácii, ktoré sú potrebné,

aby sa jedno slovo pretransformovalo na druhé. Medzi tieto operácie patrí vkladanie, mazanie a substitúcia znakov.

		0	1	2	3	4	5
			P	**E**	**T**	**E**	**R**
0		**0**	1	2	3	4	5
1	**P**	1	**0**	1	2	3	4
2	**E**	2	1	**0**	1	2	3
3	**T**	3	2	1	**0**	1	2
4	**R**	4	3	2	1	**1**	2
5	**A**	5	4	3	2	2	**2**

Tabuľka 2.1: Príklad Levenshteinovej editačnej vzdialenosti

Na tabuľke 2.1 môžme vidieť výpočet Levenshteinovej vzdialenosti, príklad je prebratý z knihy [5]. Zvýraznené čísla znázorňujú cestu výpočtu k finálnemu výsledku. Spodný pravý roh matice určuje vzdialenosť medzi slovami. V našom prípade majú mená $s_1 = Peter$ a $s_2 = Petra$ vzdialenosť 2.

Pomocou Levenshteinovej vzdialenosti $dist_{levenshtein}$ a maxima max z dĺžok reťazcov $|s|$ vieme vypočítať podobnosť slov.

$$sim_{levenshtein}(s_1, s_2) = 1.0 - \frac{dist_{levenshtein}(s_1, s_2)}{max(|s_1|, |s_2|)}$$

Čím je hodnota bližšie 1 tým sú reťazce podobnejšie a naopak, ak sa hodnota rovná 0 sú úplne rozdielne.

$$sim_{levenshtein}(s_1, s_2) = \begin{cases} 1.0 & ak\ s_1 = s_2 \\ 0.0 & ak\ s_1 \neq s_2. \end{cases}$$

Pre nami zvolené mená $s_1 = Peter$ a $s_2 = Petra$ by sme vypočítali podobnosť nasledovne $sim_{levenshtein}(s_1, s_2) = 1.0 - \frac{2}{max(|5|,|5|)}$. Výsledok by bol $sim_{levenshtein}(s_1, s_2) = 0.6$, ktorý znamená čiastočnú podobnosť.

Porovnávanie dátumov

Väčšina porovnávaných údajov je založená na porovnávaní slov, respektíve reťazca písmen. Avšak pri genealogických dátach je dôležitým faktorom aj dátum. Ten môžme brať tiež ako slovo. Je teda možné, z neho vytvoriť reťazec čísel a porovnať ho Levenshteinovou vzdialenosťou. Iný prístup je porovnávať dátumy ako špeciálne číselné údaje.

V prípade matričného záznamu o manželstve sa často uvádzal vek partnerov vstupujúcich do manželstva. Budeme musieť teda prepočítavať dátumy narodenia na vek v dni konania sobáša. Nemôžme však zabudnúť na istú toleranciu, keďže v minulosti si ľudia často krát presne nepamätali, koľko mali rokov. Túto toleranciu si určíme v percentách ako apc_{max}. Následne si vypočítam vekový rozdiel v percentách apc (age percentage diffetence), hodnoty veku sú označené ako d_1 a d_2:

$$apc = \frac{|d_1 - d_2|}{max(|d_1|, |d_2|)} \cdot 100$$

Na základe rozdielu si vypočítame podobnosť veku.

$$sim_{age} = \begin{cases} 1.0 - \frac{apc}{apc_{max}} & ak \quad apc < apc_{max} \\ 0.0 & inak. \end{cases}$$

Špeciálnou chybou u dátumov je prehodenie mesiaca a dňa. Pri dátumoch v ktorých je mesiac nad číslo 12 a deň do tohto čísla môžme predpokladať prehodenie týchto údajov. Teda napríklad u dátumov [25.1.1900] a [1.25.1900] môžme predpokladať podobnosť $sim_{date} = 0.5$.

Porovnávanie geografických dát

Okrem porovnávania podobnosti názvu miest môžeme použiť aj ďalší spôsob výpočtu podobnosti na základe porovnávania vzdialenosti medzi miestami. V minulosti nebolo tak bežné sťahovanie. Preto môžme predpokladať, že čím bližšie pri sebe sú údaje o tom istom človeku, tým pravdepodobnejšie to bude on. Prípadne, ak mal svadbu, alebo pohreb v inom blízkom meste, ako bolo jeho rodisko. Geografická vzdialenosť sa meria pozdĺž zemského povrchu v kilometroch. V prípadoch, keď chýba údaj o adrese, alebo je iba čiastočný môže sa nám podariť zaradiť, iba región z ktorého daná osoba pochádza. Potom môžme vzdialenosť počítať iba na základe stredu regiónu. Musíme však zohľadniť fakt, že výsledná vzdialenosť je len približný údaj a má preto nižšiu váhu pri porovnávaní.

Porovnávanie čísel

Pri genealogických záznamoch máme iba jeden typ číselného porovnávania a to je číslo ulice. Ideálny prípad porovnávania by bolo na základe celej adresy vyhľadanie v katastri a podľa vzdialenosti určiť podobnosť. V mestách malá chyba nemusela znamenať takú vzdialenosť ako na dedine. Tu však narážame na problém s prečíslovávaním a premenovaním celých dedín či ulíc. Preto tento prístup nie je vhodný. Následne môžme spočítať vzdialenosť čísel numericky.

$$sim_{num} = \begin{cases} 1.0 - \frac{|n_1 - n_2|}{dmax} & ak \quad |n_1 - n_2| < d_{max} \\ 0.0 & inak. \end{cases}$$

Kde n_1 a n_2 sú čísla a d_{max} je tolerovaný rozdiel medzi nimi.

V tomto prípade by však čísla 9 a 11 mali vyššiu podobnosť ako 99 a 89, ktoré sa na seba viac podobajú. Preto bude potrebné spojiť porovnávanie čísel a reťazcov, a porovnávať číslo ulice obidvoma spôsobmi.

Porovnávanie záznamov

Každý záznam sa skladá z niekoľkých atribútov rôznych typov. Každý z nich treba podrobnejšie porovnať. Z nich nám vznikne vektor hodnôt. Tu je dôležité, aby sme nebrali ako kritérium iba sčítanie týchto hodnôt, lebo by to mohlo viesť k zavádzajúcim výsledkom.

Meno, priezvisko, pohlavie a mesto sú hodnoty, ktoré prešli indexáciou a teda budeme ich brať buď ako úplnú zhodu alebo nezhodu. Následne dátum narodenia sme v tomto prípade porovnávali ako reťazec iba s jednou chybou. Podobnosť názvu ulíc sme vypočítali pomocou Levenshteinovej vzdialenosti. Pri čísle domu vidíme, že nemajú ani jedno číslo rovnaké a ani nie sú blízko pri sebe, takže sa vôbec nepodobajú. Pri mestách sme určili podobnosť reťazcov, ktorá nám vyšla 0. Na základe vektoru, ktorý vznikne bude prebiehať klasifikácia.

ID	Meno	Priezvisko	Pohlavie	Dátum narodenia	Č. domu	Názov ulice	Mesto
a1	alica	mlynárová	žena	18.10.1956	15	božetechova	brno
a2	alica	mikuláková	žena	18.10.1958	68	dožetekova	břeclav
	1.0	0.0	1.0	0.87	0.0	0.73	0.0

Tabuľka 2.2: Porovnanie dvoch osôb

2.2.5 Klasifikácia

Pred tým ako sa rozhodneme, že prepojíme dve osoby do vzťahu musíme vedieť, že v dvoch záznamoch figuruje tá istá osoba. Základný princíp je jednoduchý, čím viac sú si dva záznamy podobné, tým je pravdepodobnejšie, že pôjde o tú istú osobu. Keďže nám chýba jedinečný identifikátor osoby, ktorý je v súčastnosti rodné číslo, je potrebné na porovnávanie záznamov použiť dostupné atribúty.

Budeme pracovať s databázou, ktorá sa vzťahuje na určitú populáciu. Osoby v matričných záznamoch sa však budú čiastočne prekrývať. Informácie v matričných záznam rozdelíme na jednotlivé osoby, ktoré budeme medzi sebou porovnávať. Predpokladáme, že každý záznam o osobe sa vzťahuje na jednotlivca. Dvojice jednotlivých porovnávaných osôb z databáz budeme označovať ako r_i a r_j.

Pri porovnávaní záznamov, čo bolo rozobraté v predchádzajúcej časti 2.2.4 nám vznikne porovnávací vektor γ. Tento vektor je vygenerovaný pre každý pár. Po sčítaní všetkých hodnôt v ňom nám vyjde celková zhoda $SimSum$. Na základe tejto hodnoty budeme záznamy rozdeľovať do 3 kategórií a to zhody, potencionálne zhody a nezhody.

$$SimSum[r_i, r_j] \geq t_u \implies r \to Zhoda,$$
$$t_l < SimSum[r_i, r_j] < t_u \implies r \to Potencionlna \quad zhoda,$$
$$SimSum[r_i, r_j] \leq t_l \implies r \to Nezhoda$$

Hodnoty prahov t_l a t_u sú teda jedným z kľúčových aspektov porovnávania. Dajú sa nastaviť na pevné hodnoty, alebo ich môžme zistiť testovaním. Našou úlohou bude nájsť správne hodnoty prahov t_u a t_l. Budeme sa snažiť maximalizovať množstvo správne klasifikovaných hodnôt.

Všetky výsledky porovnávania sú normalizované medzi 0 a 1, takže všetky atribúty prispievajú rovnakým spôsobom ku konečnej súčtovej hodnote podobnosti. Dôležitosť rôznych atribútov, je teda zanedbaná. Tento nedostatok sa odstránime váženými hodnotami podobnosti, pričom rôzne atribúty dostanú rôzne váhy podľa dôležitosti. Váhy budeme určovať na základe informácie, ktorú nesú. Napríklad, ak ide o atribút priezvisko, váha tohto faktu by mala byť vyššia, ako váha povolania, pretože tých môže mať za život človek viac. Vážená suma sa vypočíta tak, že sa pred súčtom sa vynásobia hodnoty podobnosti váhou podľa typu atribútu.

Napríklad za predpokladu, že k atribútom v tabuľke 2.2 sú priradené nasledujúce váhy w:

Vzhľadom na $w_{Meno} = 2$, $w_{Priezvisko} = 3$, $w_{C.domu} = 1$, $w_{Ulica} = 3$, $w_{Mesto} = 2$, $w_{Datum\_narodenia} = 3$.

Vážená podobnosť pre dva danú dvojicu osôb je potom:

$$SimSum[a1, a2] = 2 \times 1.0 + 3 \times 0.0 + 1 \times 1.0 + 3 \times 0.87 + 1 \times 0.0 + 3 \times 0.73 + 2 \times 0.0 = 7.8$$

Pravdepodobnostná klasifikácia

Pravdepodobnostné klasifikovanie je tradičný princíp, ktorý bol predstavený Ivanom Fellegi a Alanom Sunterom v roku 1969 [9]. Ich práca a Theory For record Linkage tiež anglicky známa ako 'probabilistic record linkage', v slovenskom jazyku pravdepodobnostné prepojenie záznamu, sa stala základom mnohých systémov na porovnávanie záznamov.

Váhy týchto atribútov by však nemali závisieť len od všeobecných charakteristík atribútov, ale aj od ich skutočných hodnôt. Napríklad, ak ide o atribút priezvisko, ak sa dvaja ľudia volajú Novák, váha tohto faktu by nemala byť veľmi vysoká. Ide o najčastejšie české priezvisko a preto je vysoká pravdepodobnosť, že náhodne vybraní ľudia z databáze budú mať práve toto priezvisko. Naopak, ak majú dva záznamy priezvisko Višek, ktorých sa na území Českej republiky nachádza len 9, mala by váha takéhoto faktu byť výrazne väčšia. Následne si formálne zadefinujeme, ako bude toto prepájanie záznamov fungovať. Ako aj v predchádzajúcom prípade, budeme záznamy rozdeľovať do 3 kategórií a to zhody, potencionálne zhody a nezhody.

Budeme pracovať s databázou, ktorá sa vzťahuje na určitú populáciu A a B. Teda sa čiastočne prekrývajú. Predpokladáme, že každý záznam sa vzťahuje na jednotlivca. Jednotlivé záznamy z týchto databáz budeme označovať ako a a b.

$$a \times B = \{(a, b) : a \in A, b \in B\}$$

Táto množina bude pozostávať z dvoch disjunktných množín. Prvá bude množina zhôd M (anglicky matches). Teda záznamy a a b odpovedajú tej istej osobe. Predpokladáme, že môže ísť o tú istú osobu v prípade, ak sa záznamy úplne zhodujú, ale rovnako aj ak sa v niektorých hodnotách líšia

$$M = \{(a, b) : a = b, a \in A, b \in B\}$$

Druhá bude množina nezhôd U (anglicky non-matches). Osoby a a b sú rôzne.

$$U = \{(a, b) : a \neq b, a \in A, b \in B\}$$

Pri porovnávaní záznamov, čo bolo rozobraté v predchádzajúcej časti 2.2.4 nám vznikne porovnávací vektor γ. Tento vektor je vygenerovaný pre každý pár. V základnej formulácii sú brané v úvahu iba binárne hodnoty. Pravdepodobnosť zhody potom vypočítame, ako podmienenú pravdepodobnosť zhodných a nezhodných údajov v zázname.

$$R = \frac{P(\gamma \in \Gamma | r \in M)}{P(\gamma \in \Gamma | r \in U)}$$

Na základe tejto hodnoty sa rozhodujeme, či pôjde o zhodu alebo nie.

$$R \geq t_u \implies r \to Zhoda,$$
$$t_l < R < t_u \implies r \to Potencionlna \quad zhoda,$$
$$R \leq t_u \implies r \to Nezhoda$$

Z týchto pravidiel vyplýva, že čím viac podobných hodnôt pre danú osobu existuje, tým je väčšia pravdepodobnosť, že pôjde o tú istú osobu. Na druhú stranu čím viac bude nezhôd tým menšia bude pravdepodobnosť, že ide o rovnakú osobu.

Výpočet podmienených pravdepodobností je hlavným bodom pravdepodobnostného prepojenia záznamov. Všeobecne sa predpokladá, že tieto pravdepodobnosti sú nezávislé

pre rôzne atribúty. Za tohto predpokladu je možné, pre každý atribút vypočítať individuálnu váhu w_i, pre každý atribút i, na základe pravdepodobností m a u. Pričom sa porovnávajú dve hodnoty atribútu a_i a b_i.

$$m_i = P([a_i = b_i, a \ \in A, b \in B]|r \in M)$$
$$u_i = P([a_i = b_i, a \ \in A, b \in B]|r \in U),$$

Pravdepodobnosť m_i určuje, že dva záznamy majú rovnakú hodnotu v atribúte i, pretože daný pár je zhoda. Na druhej strane, u_i je pravdepodobnosť, že dva záznamy majú rovnakú hodnotu v atribúte i,napriek tomu, že daný pár je nezhoda.

Individuálna váha w_i pre atribút i, sa vypočíta na základe týchto dvoch pravdepodobností ako:

$$w_i = \begin{cases} log_2 \frac{m_i}{u_i} & ak \quad a_i = b_i, \\ log_2 \frac{1-m_i}{1-u_i} & a_i \neq b_i \end{cases}$$

Týmto spôsobom, sme schopní vypočítať váhu, pre každý atribút.

2.2.6 Vyhodnotenie výsledkov

Posledným krokom tejto práce je vyhodnotenie správnosti prepájania záznamov. K tomu nám budú slúžiť testovacie dáta. Na základe toho, že pri týchto dátach vieme, ako sú v skutočnosti prepojené a budeme vedieť kontrolovať klasifikáciu. Každé porovnanie teda skontrolujeme a priradíme do jednej zo skupín:

- Pravdivá zhoda (angl. True positive) - TP - záznamy o osobe, odkazujú na tú istú osobu a klasifikovali sme ich ako zhodné, klasifikovali sme ich správne
- Nepravdivá zhoda (angl. False positive) - FP - záznamy o osobe, odkazujú na rôzne osoby a klasifikovali sme ich ako zhodné, klasifikovali sme ich nesprávne
- Pravdivá nezhoda (angl. True negative) - TN - záznamy o osobe, odkazujú na rôzne osoby a klasifikovali sme ich ako rozdielne, klasifikovali sme ich správne
- Nepravdivá nezhoda (angl. False negative) - FN - záznamy o osobe, odkazujú na tú istú osobu a klasifikovali sme ich ako rozdielne, klasifikovali sme ich nesprávne

Tieto hodnoty sa často zaznamenávajú v chybovej matici 2.3.

		Predpovedaný výsledok	
		Zhoda	Nezhoda
Skuočný výsledok	Zhoda	Pravdivá zhoda - TP	True negative - TN
	Nezhoda	Nepravdivá zhoda - FP	False negative - FN

Tabuľka 2.3: Chybová matica

Snažíme sa o to, aby sme čo najviac výsledkov predpovedali správne a čo najmenej nesprávne. Teda aby skupiny TP a TN boli čo najväčšie a naopak FN a FP čo najmenšie. Na základe týchto údajov sa dajú vypočítať hodnoty, ktoré určujú vlastnosti a vhodnosť kvality porovnávania.

Presnosť

Presnosť alebo v anglickom jazyku precision sa počíta nasledovne:

$$precision = \frac{TP}{TP + FP}$$

Ide o pomer správne klasifikovaných zhôd a všetkých hodnôt, čo sme klasifikovali ako zhodné.

Citlivosť

Citlivosť alebo v anglickom jazyku recall sa počíta nasledovne:

$$recall = \frac{TP}{TP + FN}$$

Výsledkom je teda pomer správne klasifikovaných zhôd a všetkých hodnôt, čo sú naozaj zhodné.

F-miera

F-miera alebo v anglickom jazyku F-measure sa počíta nasledovne:

$$F - measure = 2 \times \frac{(precision \times recall)}{precision + recall}$$

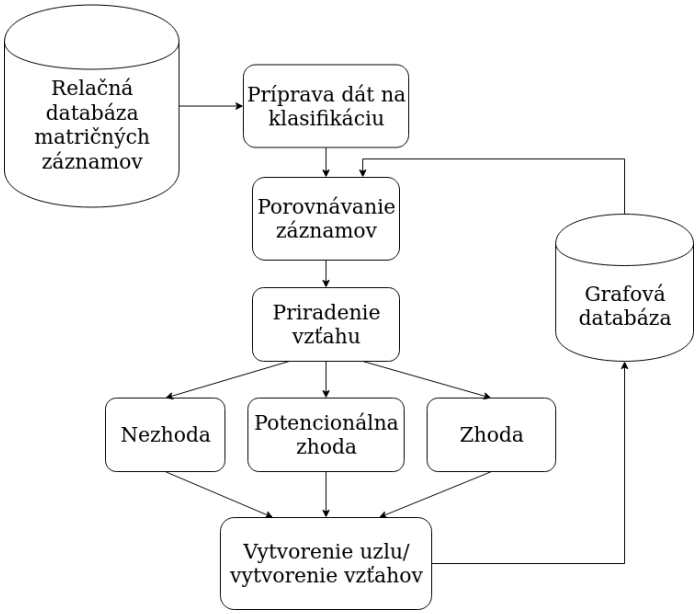
Hodnota sa určuje, ako harmonický priemer dvoch predchádzajúcich metrík.

Kapitola 3

Návrh riešenia

Samotnej implementácii systému predchádza jeho návrh. V nasledujúcej kapitole si priblížime, ako budeme postupovať pri realizácii na základe nadobudnutých teoretických znalostí.

Na obrázku 3.1 môžme vidieť grafické znázornenie navrhovaného systému.



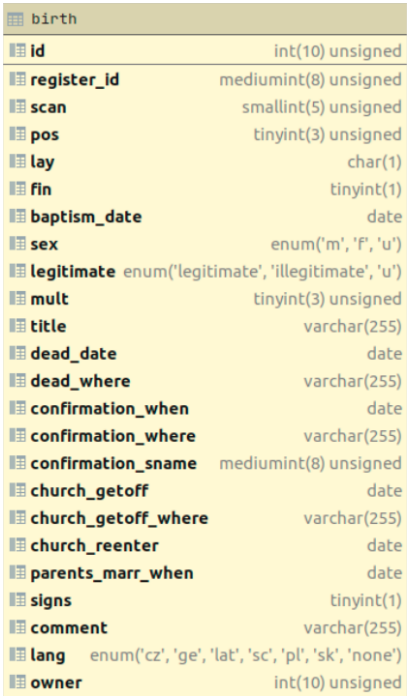
Obr. 3.1: Návrh prepojovania záznamov

Prvý krok je spracovanie relačnej databáze načítanie dát do vnútornej reprezentácie, aby sme vedeli ďalej pracovať so záznamom a osobami v ňom. Ďalšia časť je grafová databáza, z ktorej si rovnako budeme musieť spracovávať dáta. Okrem toho na rozdiel od relačnej si grafovú databázu aj celú navrhneme. Po tom čo spracujeme dáta z obidvoch databáz prichádza na rad porovnávanie záznamov, ktoré je hlavnou časťou tejto práce. Na základe výsledku porovnávania následne rozhodneme, či budeme vytvárať nový uzol, alebo nie. Ak je výsledok porovnávania negatívny, teda sme nenašli zhodu vytvoríme nový samostatný uzol. Ak sme našli pri porovnávaní zhodu, rozšírime informácie v už existujúcom uzli. V prípade potencionálnej zhody vytvoríme nový uzol a prepojíme ho s už existujúcim. Môžeme rozdeliť riešenie do 3 logických celkov a to spracovanie relačnej databázy, práca s grafovou databázou a práca so záznamom. Tieto časti systému si v nasledujúcich kapitolách podrobne rozoberieme.

3.1 Relačná databáze

Budeme pracovať s reálnymi dátami, ktoré sú uložené už v existujúcej databáze **perun**. Tá sa nachádza na školskom serveri *fit.vutbr.cz*. Jedná sa o MySQL relačnú databázu, ktorú postupne celú spracujeme. Pod relačnou databázou si môžme predstaviť dáta, ktoré sú uložené v tabuľkách. Tieto tabuľky sú navzájom prepájané väzbami. K záznamom budeme pristupovať jednotlivo a pomocou dotazov, budeme vyberať postupne všetky prepojené údaje a pracovať s nimi. Následne sa nám táto databáza môže zväčšovať a bude potrebné postupne spracovávať nové záznamy. V každom zázname bude uložená informácia o tom, či už bol spracovaný, alebo nie. Preto vždy pred načítaním kompletného záznamu a všetkých náležitých dát skontrolujem tento príznak.

V čase práce na diplomovej práci som mala k dispozícii iba databázu matrík narodených. Celý ER diagram databáze sa nachádza v prílohe B, keďže je príliš veľký a neprehľadný postupne si rozoberieme pre nás podstatné časti. Najdôležitejšia je tabuľka *birth*, ktorú môžme vidieť na obrázku 3.2. Je nositeľom informácii o matričnom zázname a o osobe, ktorá bola krstená.

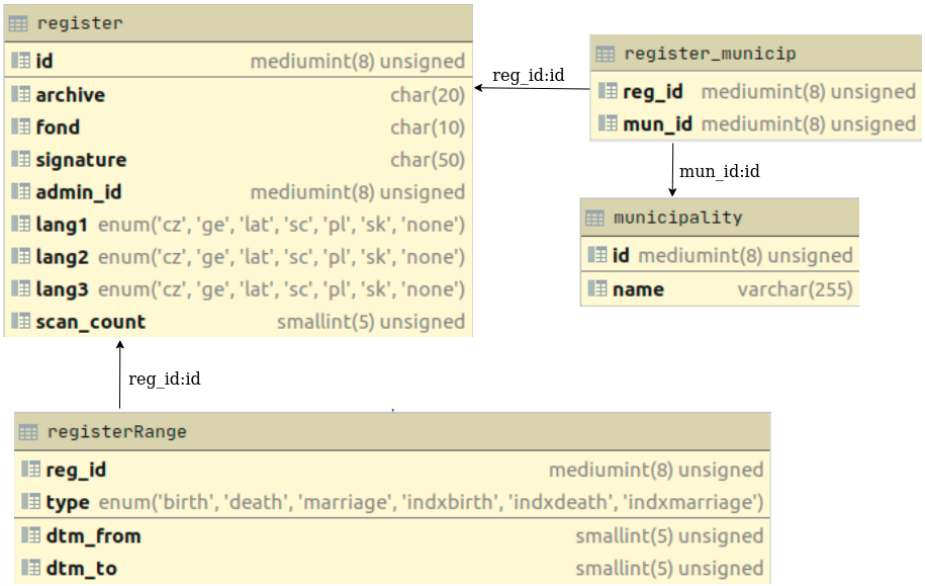


Obr. 3.2: Tabuľka birth z relačnej databázy

V tabuľke *birth* máme informácie o matričnom zázname. Keďže názvy premenných sú zapísané pomocou skratiek, vysvetlíme si čo znamenajú. *Scan* určuje poradie skenu, *lay* rozloženie na skene, *pos* poradie záznamu a *lang* jazyk v ktorom je záznam napísaný. Ďalšie informácie, ktoré sa nachádzajú v tejto tabuľke sa týkajú krstňaťa. Nachádzajú sa tu informácie o dátume krstu, pohlaví, titule, mieste a dátume úmrtia, birmovky prípadne o odchode či znova vstúpení do cirkvi. Ďalej je tu údaj o tom, či ide o viacerčatá, či je to manželské alebo nemanželské dieťa, alebo o dátume svadby rodičov prípadne, či bol otec v čase narodenia mŕtvy. Avšak to neznamená, že všetky tieto informácie skutočne máme. Vo väčšine prípadov sú tieto informácie nevyplnené.

3.1.1 Matrika

Vzhľadom na to, že pracujeme so záznamami z rôznych matrík, potrebujeme spracovať aj informácie o nich. V týchto tabuľkách, ktoré sú na obrázku 3.3 sa nachádzajú všetky údaje o matrike. Pričom hlavná tabuľka nesie názov *register* v nej sa nachádzajú informácie o archíve, fonde, signatúre jazykoch a množstve skenov. Tabuľka *registerRange* zaznamenáva časové obdobie, kedy bola matrika používaná. Posledná tabuľka *municipatily* nesie údaje s názvom okresu. Tabuľka *register_municip* nám pomáha iba ako prepojenie tabuliek *register* a *municipality*. Ak chceme mať kompletné informácie o osobe, chceme vedieť aj v ktorých matrikách bola zapísaná. Každá matrika môže mať logicky mnoho záznamov teda vzťah tabuľky *register* a *birth* bude n:1.



Obr. 3.3: Všetky tabuľky z relačnej databázy, ktoré obsahujú podstatné informácie o matrike

3.1.2 Osoba

Na získanie všetkých informácii o osobe potrebujeme až 13 tabuliek. Je to z toho dôvodu, že dáta ktoré sú normalizované, sú uložené v samostatných tabuľkách. Na prípravu dát a ich normalizáciu je využitá bakalárska práca študenta Davida Hříbeka s názvom Poloautomatická normalizace slov z matričních záznamů [10]. Na základe tejto práce sa využitím metódy zhlukovania normalizujú krstné mená, priezviská, názvy miest, povolania a vzťahy medzi osobami.

Pre každú osobu zo záznamu je vytvorená tabuľka *birthPerson*. Tá je prepojená s tabuľkou *birth* vzťahom 1:n. Jeden záznam obsahuje mnoho osôb. V tabuľke *birthPerson* sú uložené nasledujúce dáta, titul, ulica bydliska, popisné číslo, náboženstvo, dátum narodenia a úmrtia. Nachádza sa tu ešte údaj o vzťahu respektíve o role osoby v zázname. Vieme podľa toho či šlo o otca, matku, kňaza a podobne. Pričom je tento údaj uložený aj v slovnej podobe v inej tabuľke, ale aj vybratím jednej z možnosti z rolí. Ak nie je pri osobe pohlavie definované môžme sa ho pokúsiť zistiť na základe role. Pri otcovi alebo matke vieme pohlavie ľahko určiť. Nie vždy sa to však dá napríklad pri roli *KMOTR* nevieme či ide o ženu

alebo muža. Všetky existujúce role si rozoberieme podrobnejšie v časti o grafovej databáze 3.2.1.

Tabuľka *birth* je vytvorená aj pre krstňa. Pričom sa tu ešte nachádzajú dva údaje, ktoré sa týkajú iba neho a to sú príznaky o tom či sa narodilo mŕtve, alebo či ide o nalezenca. Pričom krstňa, osoba ktorá bola krstená, má hlavnú úlohu, ktorá sa nazýva *main*.

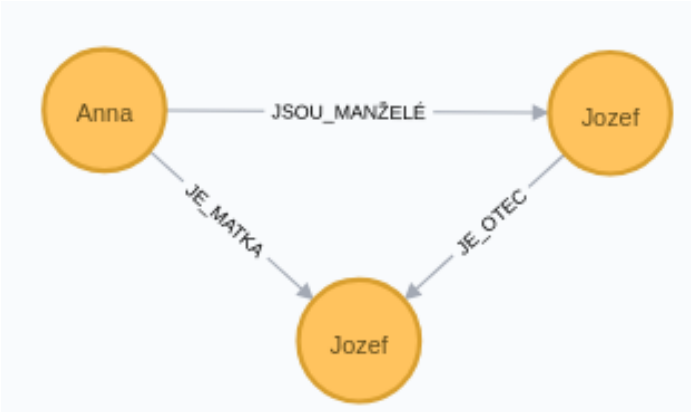
Obr. 3.4: Všetky tabuľky z relačnej databázy, ktoré obsahujú podstatné informácie o osobe zo záznamu

Z obrázka 3.4 si môžme všimnúť, že údaje na ktoré bola použitá metóda zhlukovania sa nenachádzajú v hlavnej tabuľke, ale majú vytvorené ďalšie tabuľky. Ak chceme teda napríklad priezvisko osoby, ktorej sa týka tabuľka *birthPerson* musíme prejsť do tabuľky *surname*. V tejto tabuľke je uložené aj pohlavie k akému priezvisko patrí. Normalizovaná forma priezviska je až v ďalšej tabuľke s názvom *normalizedSurname*. Rovnako ak chceme zistiť názov mesta a jeho normalizovanú formu pristupujeme k tabuľkám *domicile* a *normalizedDomicile*. To isté platí aj pre rolu osoby v zázname a jej normalizovanú formu, ktoré sa nachádzajú v tabuľkách *personRelation* a *normalizedPersonRelation*. Okrem týchto in-

formácii sa už v spomenutých tabuľkách nenachádzajú žiadne ďalšie podstatné informácie. S menom a povolaním je to o čosi komplikovanejšie, keďže k nim pristupujeme ešte cez jednu tabuľku. Pričom tabuľky *birthPerson_name* a *birthPerson_occup* tvoria len prepojenie na ďalšie tabuľky a nenesú žiadne dáta. Tak ako pri priezvisku aj krstné meno má v tabuľke *name* uložené pohlavie. V prípade ak by osoba nemala zadané pohlavie, môžme ho skúsiť zistiť z týchto tabuliek. Normalizované meno potom rovnako ako v predošlých prípadoch sa nachádza v samostatnej tabuľke *normalizedName*. V tabuľkách *occupation* a *normalized_occupation* sa nachádzajú údaje o zamestnaní a jeho normalizovanej forme. Z všetkých týchto údajov si postupne vytvoríme osobu s ktorou budeme ďalej pracovať.

3.2 Grafová databáza

Tento typ databázy ukladá dáta výrazne odlišne ako relačná databáza. Dáta sú ukladané v grafovej štruktúre. Uzly sú nositeľmi informácii a hrany určujú vzťahy medzi nimi. Je podstatné, aby sme pri prevode z relačnej na grafovú databázu, nestratili žiadne dôležité informácie. Tieto databázy sa využívajú na ukladanie dát medzi, ktorými sa nachádzajú určité vzťahy. Príkladom môžu byť zamestnanci ktorí, pracujú na rovnakých projektoch v rôznych mestách. Prípadne herci, ktorí hrajú v rôznych filmoch pre rôzne štúdia. Ďalším typickým príkladom sú práve rodokmene. Pre nás je najdôležitejší aspekt vytvorenie vzťahu medzi jednotlivými osobami. Príklad základnej väzby medzi rodičmi a dieťaťom v grafovej databáze môžme vidieť na obrázku 3.5.



Obr. 3.5: Príklad základných vzťahov medzi osobami v grafovej databáze

3.2.1 Návrh databáze

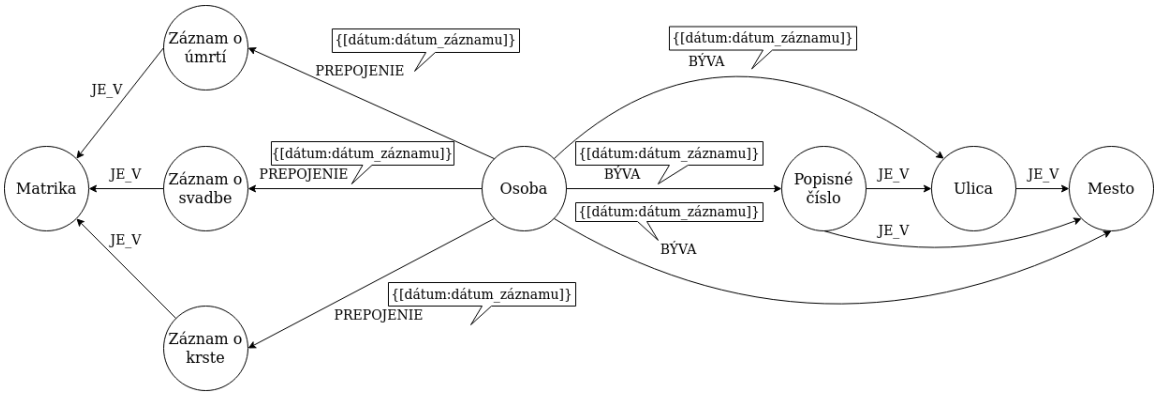
V úvode by som chcela upozorniť, že i keď je táto diplomová práca písaná v slovenskom jazyku databáza je navrhnutá v českom jazyku. Preto budú názvy uzlov a rovnako aj názvy premenných a vzťahov v češtine.

Na obrázku 3.6 môžme vidieť návrh databázy. Bude existovať 8 typov uzlov:

- Matrika
- Záznam o krste
- Záznam o manželstve
- Záznam o úmrtí
- Osoba
- Popisné číslo

- Ulica
- Mesto

Pričom všetky dáta o danom uzle budú uložené v ňom. Pre ponechanie prehľadnosti obrázka 3.6, som vzťahy medzi záznamami a osobou nazvala PREPOJENÍM v skutočnosti však toto prepojenie určuje rolu osoby v zázname.



Obr. 3.6: Návrh prepojovania záznamov

Z obrázka 3.6 teda vidíme, že z jednej osoby vychádza viac šípok prepojenie. To značí, že sa môže nachádzať vo viacerých záznamoch. Rovnako môže počas života bývať na viacerých miestach. Keďže matričné záznamy nemusia obsahovať celú adresu, ale iba jej časť prípadne iba mesto, tak osobu naviažeme na najdetailnejšiu informáciu. Následne sú hierarchicky prepojené ďalšie časti adresy. Vzťahu BÝVA je priradená vlastnosť dátum. Nachádza sa tam z dôvodu spätného prepojenia. Podľa dátumu budeme vedieť prepojiť záznam s adresou. Budeme vedieť zistiť, kde osoba žila v ktorom období života. Rovnako aj vzťah PREPOJENIE má vlastnosť dátum z rovnakého dôvodu. V prípade ak by napríklad osoba mala viac vzťahov OTEC na základe dátumu budeme vedieť, kedy mala ktoré dieťa. Záznamy sú prepojené na matriku v ktorej sa nachádzajú.

Osoba	Vzťah	Záznam
Osoba	KŘTENEC	Záznam_o_křte
Osoba	OTEC	Záznam_o_křte
Osoba	MATKA	Záznam_o_křte
Osoba	POROBNÍ_BÁBA	Záznam_o_křte
Osoba	KŘTITEL	Záznam_o_křte
Osoba	OTCŮV_OTEC	Záznam_o_křte
Osoba	OTCOVA_MATKA	Záznam_o_křte
Osoba	MATČIN_OTEC	Záznam_o_křte
Osoba	MATČINA_MATKA	Záznam_o_křte
Osoba	OTEC_MATČINHO_OTCA	Záznam_o_křte
Osoba	OTEC_MATČINY_MATKY	Záznam_o_křte
Osoba	MATKA_MATČINHO_OTCA	Záznam_o_křte
Osoba	MATKA_MATČINY_MATKY	Záznam_o_křte
Osoba	KMOTR	Záznam_o_křte
Osoba	PRIBUZDNÝ_KMOTRA	Záznam_o_křte
Osoba	MANŽEL_MATKY	Záznam_o_křte

Tabuľka 3.1: Vzťahy medzi osobami v zázname o krste

Keď sme si už definovali základnú štruktúru databázy rozoberieme všetky vzťahy, ktoré sa nachádzajú medzi osobou a záznamami. Všetky tieto role sú vyjadrené vzťahmi, ktoré môžme vidieť v tabuľke 3.1.

V zázname o krste vystupujú tradičné postavy ako otec, matka, krstňa či krstiteľ. Okrem starých rodičov dieťaťa sa uvádzali aj rodičia starých mám. Žena bola v záznamoch často upresnená svojím mužským príbuzným, keďže menila počas života priezvisko. Dieťa mohlo mať aj niekoľko krstných rodičov pričom opätovne, ak bola uvedená žena, bývala väčšinou upresnená mužským príbuzným. V prípade ak ženin manžel nebol otcom dieťaťa, uvádzal sa aj skutočný otec dieťaťa. To znamená, že pri dieťati bola uvedená matka, jej manžel aj biologický otec dieťaťa.

Databázy so záznamami o manželstve a úmrtí neboli v čase vypracovania diplomovej práce ešte pripravené. Mala som, však prístup k tomu aké dáta v nich budú uložené, preto som vypracovala návrh databáze. Tieto databázy budú spracované v budúcnosti. Navýšia počet možných vzťahov a rozšíria množstvo informácií o osobe.

Záznam o manželstve je určite najobsiahlejší, čo sa týka množstva rolí, ktoré sa v ňom môžu vyskytovať. Uvedení sú rodičia a prarodičia obidvoch zúčastnených, teda nevesty aj ženícha, čo výrazne zvyšuje počet osôb. Ak vstupoval do manželstva vdovec alebo vdova, je uvedený aj predchádzajúci manžel alebo manželka. V prípade ak ženích alebo nevesta už nemali živých rodičov, ale mali opatrovníka, tak sa uvádzal aj ten. Okrem toho tu sú role typické pre svadbu ako svedok, rečník, stará, družba a družička. Príbuzný svedka sa uvádzal z rovnakého dôvodu ako príbuzný kmotra, teda ak šlo o ženu, aby bola lepšie identifikovateľná.

Osoba	Vzťah	Záznam
Osoba	ODDÁVAJÍCÍ	Záznam_o_oddaní
Osoba	ŽENICH	Záznam_o_oddaní
Osoba	NEVĚSTA	Záznam_o_oddaní
Osoba	VDOVEC_PO	Záznam_o_oddaní
Osoba	OTEC_ŽENICHA	Záznam_o_oddaní
Osoba	MATKA_ŽENICHA	Záznam_o_oddaní
Osoba	OTEC_MATKY_ŽENICHA	Záznam_o_oddaní
Osoba	MATKA_MATKY_ŽENICHA	Záznam_o_oddaní
Osoba	OPATROVÍK_ŽENICHA	Záznam_o_oddaní
Osoba	VDOVA_PO	Záznam_o_oddaní
Osoba	OTEC_NEVĚSTY	Záznam_o_oddaní
Osoba	MATKA_NEVĚSTY	Záznam_o_oddaní
Osoba	OTEC_MATKY_NEVĚSTY	Záznam_o_oddaní
Osoba	MATKA_MATKY_NEVĚSTY	Záznam_o_oddaní
Osoba	OPATROVNÍK_NEVĚSTY	Záznam_o_oddaní
Osoba	SVĚDEK	Záznam_o_oddaní
Osoba	PŘÍBUDNÍ_SVĚDKA	Záznam_o_oddaní
Osoba	ŘEČNÍK	Záznam_o_oddaní
Osoba	STARÁ	Záznam_o_oddaní
Osoba	DRUŽBA	Záznam_o_oddaní
Osoba	DRUŽIČKA	Záznam_o_oddaní

Tabuľka 3.2: Vzťahy medzi osobami v zázname o manželstve

Záznam o úmrtí je najstručnejší záznam, ako sme už uvideli v kapitole 2.1.2. Obsahuje najmenej osôb. Definuje iba jednu osobu s jej partnerom, potomkami, rodičmi prípadne

ďalším príbuzným. Pričom matka je opätovne určená presnejšie svojimi rodičmi. Vystupujú tu však dve rozdielne role oproti predchádzajúcim záznamom. Ide o rolu POHŘBÍVAJÍCÍ, ktorá označuje kňaza, teda osobu vedúcu pohreb. Ďalšia je ZAOPATROVATEĽ ide o osobu, ktorá bola s umierajúcim v dobe umierania.

Osoba	Vzťah	Záznam
Osoba	ZAOPATROVATEĽ	Záznam_o_úmrtí
Osoba	POHŘBÍVAJÍCÍ	Záznam_o_úmrtí
Osoba	ZEMŘELÝ	Záznam_o_úmrtí
Osoba	OTEC_ZEMŘELÉHO	Záznam_o_úmrtí
Osoba	MATKA_ZEMŘELÉHO	Záznam_o_úmrtí
Osoba	OTEC_MATKY_ZEMŘELÉHO	Záznam_o_úmrtí
Osoba	MATKA_MATKY_ZEMŘELÉHO	Záznam_o_úmrtí
Osoba	MAŽELKA_ZEMŘELÉHO	Záznam_o_úmrtí
Osoba	MAŽEL_ZEMŘELÉ	Záznam_o_úmrtí
Osoba	SYN_ZEMŘELÉHO	Záznam_o_úmrtí
Osoba	DCERA_ZEMŘELÉHO	Záznam_o_úmrtí
Osoba	PRÍBUZNÝ_ZEMŘELÉHO	Záznam_o_úmrtí

Tabuľka 3.3: Vzťahy medzi osobami v zázname o úmrtí

Ďalšie vzťahy tvoria jadrom tejto práce a sú to vzťahy medzi osobami. Osoba môže mať rôzne role v rôznych záznamoch, môže vystupovať ako dieťa, svedok, krstný otec, manžel, otec. Ako postupne budeme nachádzať zhody, bude sa nám zväčšovať počet rolí, ktoré bude zastupovať jedna osoba. V nasledujúcej tabuľke 3.4 sú vypísané všetky vzťahy medzi osobami.

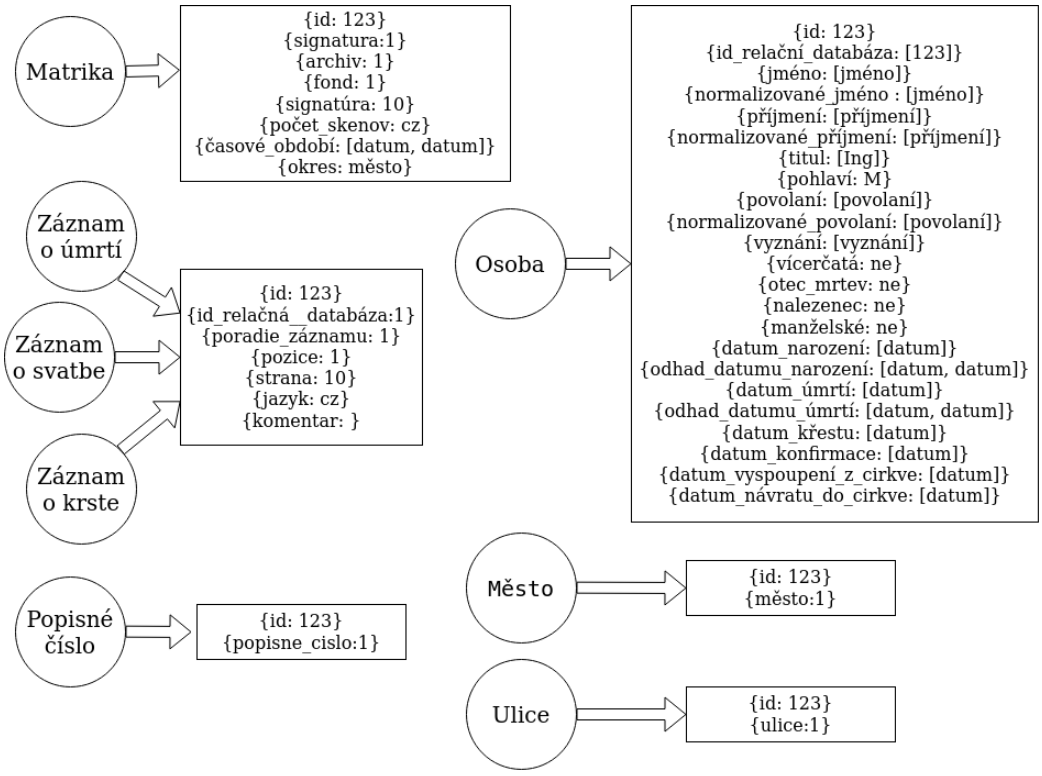
Osoba	Vzťah	Osoba
Osoba	JE_OTEC	Osoba
Osoba	JE_MATKA	Osoba
Osoba	JE_KMOTR	Osoba
Osoba	JSOU_MANŽELÉ	Osoba
Osoba	JE_SVĚDEK	Osoba
Osoba	ODRODILA	Osoba
Osoba	JE_PRÍBUZNÝ	Osoba
Osoba	ODDAL	Osoba
Osoba	POHŘBIL	Osoba
Osoba	KŘTIL	Osoba
Osoba	ODRODILA	Osoba
Osoba	ZAOPATŘIL	Osoba
Osoba	OPATROVNÍK	Osoba

Tabuľka 3.4: Vzťahy medzi osobami

Z tabuľky je na prvý pohľad jasné, že i keď je rolí mnoho vzťahov medzi ľudmi je už výrazne menej. Jeden z hlavných dôvodov je, že v grafovej databáze, sú ľudia uložený podobne ako v rodokmeni. Z toho dôvodu sú vzťahy ako napríklad MATKA_MATKY_ŽENICHA definované ako dvojitý vzťah JE_MATKA. Tento fakt zjednodušuje prácu s databázou, pretože nám okresal množstvo rolí, pod ktorými človek môže vystupovať.

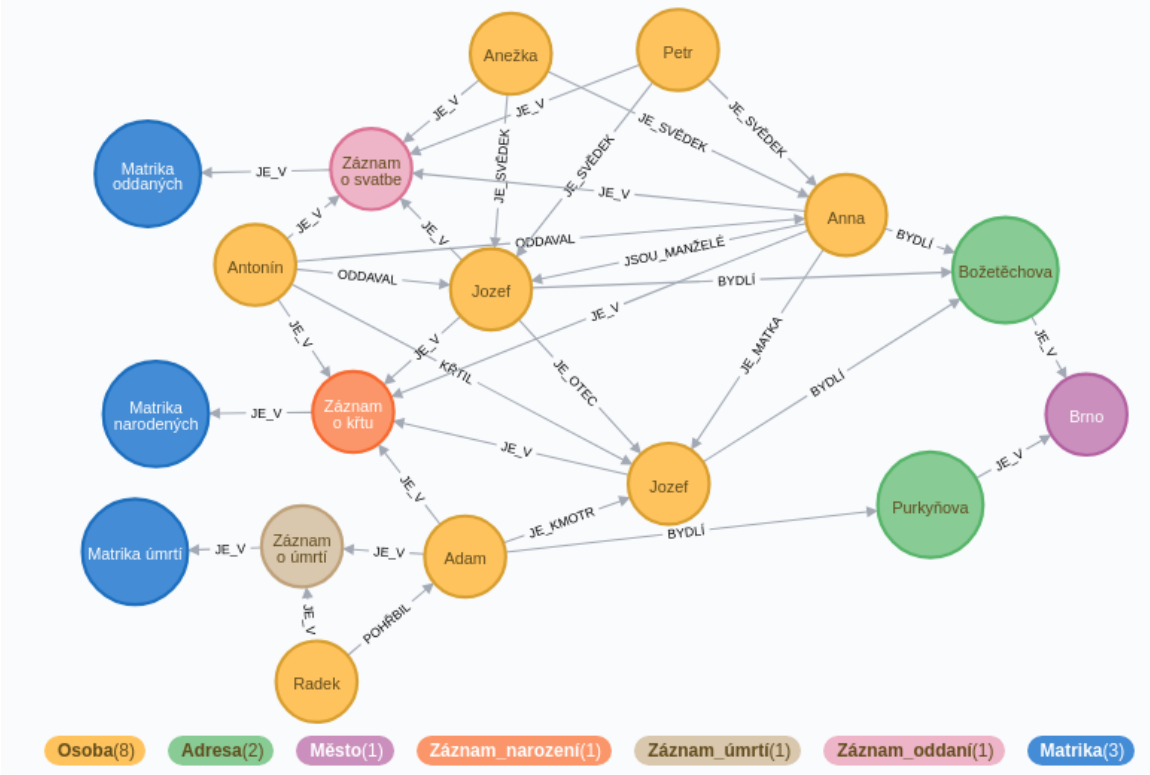
Po definovaní všetkých vzťahov sa môžme pozrieť na vlastnosti uzlov. V časti o relačnej databáze 3.1 sme uviedli, aké všetky dáta vieme získať. Na obrázku 3.7 môžme vidieť, ako

dáta ukladáme do daných uzlov. Do uzlu matrika uložíme všetky dáta, ktoré sme získali o matrike. I keď uzly záznamov sú nositeľmi rovnakých informácii je podstatné si uvedomiť, že každý z týchto uzlov môže mať rôzne vzťahy. Aby sme teda vedeli, aké vzťahy môžu byť priradené uzlu rozlišujeme typ uzla. Uzol osoba je najobšírnejší. Obsahuje všetky informácie, ktoré sme dostali z relačnej databázy. Okrem toho obsahuje aj dva dodatočné údaje a to odhad dátumu narodenia a odhad dátumu úmrtia. Všetky údaje sú uložené v zoznamoch, až na už spomínané dva dodatočne vytvorené. Tie budeme postupne aktualizovať. Pri prepájaní záznamov môže nastať situácia, že niektoré údaje sa nezhodujú. My si však budeme chcieť ponechať všetky údaje, preto si aj tieto uložíme do zoznamu. Pri osobách a záznamoch si ukladáme aj id z relačnej databázy. Budeme tak vedieť, z ktorých všetkých osôb sme poskladali uzol. V prípade nezrovnalostí vieme teda spätne vyhľadať všetky informácie v relačnej databáze. Keďže je adresa hierarchicky vytvorená, každý uzol obsahuje iba jednu informáciu.



Obr. 3.7: Uzly s vlastnosťami v grafovej databáze

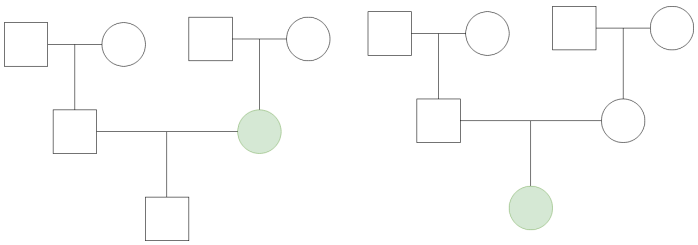
Na obrázku 3.8 vidíme názorný príklad časti databázy. Modré uzly sú matriky. Môžme vidieť, že sú prepojené len so záznamami. V príklade máme vytvorené 3 druhy záznamov každého typu. Už pri tomto relatívne malom množstve záznamov, aktívne využívame všetky tipy uzlov a použili sme množstvo druhov vzťahov. Názorne sme si ukázali, že množstvo prepojení prestáva byť prehľadné pri 3 záznamoch. Vytvorili sme 8 uzlov typu osoba. Následne sme vytvorili záznam o svadbe, ktorý obsahuje 5 ľudí. Obdobne sme vytvorili aj záznam o krste tiež s 5 ľuďmi. Pričom každý záznam by mohol byť ešte obšírnejší, ak by obsahoval údaje o rodičoch alebo krstných rodičoch. Najjednoduchší v tomto prípade je záznam o úmrtí, ktorý obsahuje len kňaza a zosnulého.



Obr. 3.8: Príklad grafovej databázy

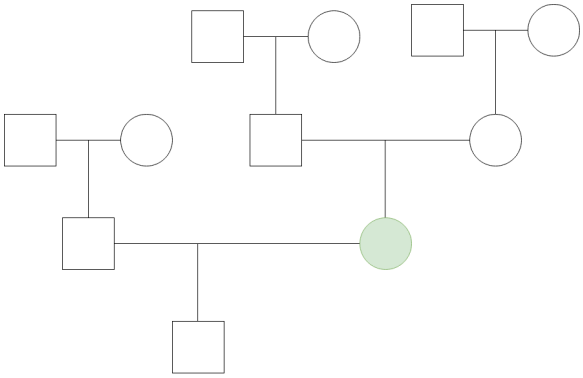
3.3 Spracovanie záznamov

Záznamy budeme porovnávať na základe pohlavia, teda mužov s mužmi a ženy so ženami. V prípade ak nebolo pohlavie určené a ani sa nám ho neporadilo získať napríklad z role osoby, tak budeme porovnávať takéto osoby vždy. Naším cieľom je prepojiť, čo najväčšie časti rodokmeňov. Idea je teda nasledovná prvým krokom je porovnanie záznamov.



Obr. 3.9: Dva rodokmene so nájdenou zhodnou osobou

V prípade ak nájdeme zhodné záznamy, spojíme ich do jedného uzlu. Následne porovnáme aj zvyšných rodinných príslušníkov. Keďže sme už našli zhodu je predpoklad, že nájdeme aj ďalšie zhody. Porovnáme preto aj ostatných rodinných príslušníkov. Týmto spôsobom párujeme časti rodokmeňov do väčších celkov.



Obr. 3.10: Prepojenie častí dvoch rodokmeňov

3.3.1 Odhady dátumov

Prvým krokom je porovnávanie. To však nemá zmysel ak vieme, že daná osoba v čase záznamu nemohla žiť. Ak napríklad záznam vznikol pred jej narodením, alebo po jeho smrti nemusíme použiť tento záznam na porovnávanie. Prvým krokom ešte pred porovnávaním ďalších hodnôt bude táto kontrola. K nej využijeme intervaly pre narodenie a úmrtie, ktoré si vygenerujeme pri načítavaní dát z relačnej databázy.

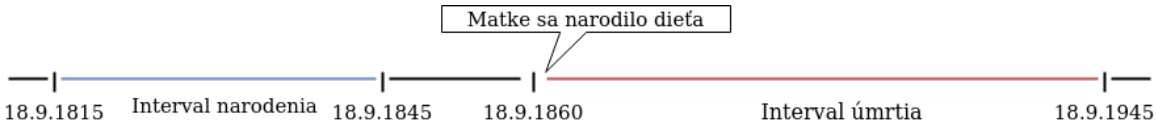
Pri odhadovaní dátumu narodenia a úmrtia sa riadime týmito časovými údajmi.

- Maximálny vek je 100 rokov
- Medzi dátumom narodenia a krstom ubehol maximálne mesiac
- Matkou alebo otcom sa mohla osoba stať najskôr v 15 rokoch
- Svadbu mohla mať osoba najskôr v 15 rokoch
- Svedok musel mat minimálne 15 rokov
- Kňazom alebo pôrodnou babou mohla byť osoba staršia ako 20 rokov
- Matkou mohla byt osoba maximálne v 45 rokov
- Otcom sa mohol stať maximálne v 55 rokov
- Konfirmáciu mohla mať osoba najskôr v 15 rokov

Pričom počítame, že osoba mohla mať svadbu, alebo konfirmáciu aj pár dní pred smrťou. Odhad pritom môžme vykonávať buď z dátumu zadaného priamo v databáze, alebo si ho môžme odvodiť z dátumu, kedy vznikol záznam. Potom na základe role, pod ktorou vystupuje daná osoba v zázname, môžme vytvoriť intervaly. V čím viac záznamoch bude osoba vystupovať, tým budeme vedieť lepšie upresniť tieto intervaly. Okrem týchto časových údajov vymenovaných vyššie existujú aj ďalšie, ktoré z nich vychádzajú. Napríklad prarodič musí mať najmenej 30 rokov, čo vychádza z veku rodiča. Ak je najmenší vek pre kňaza 20 rokov, tak sa mohol dožiť maximálne 80 rokov od dátumu záznamu. To vychádza z toho, že maximálny vek je 100 rokov.

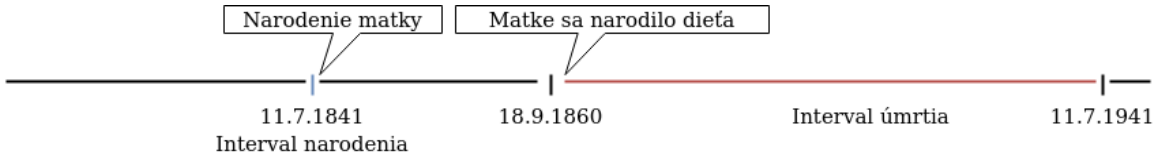
Rozoberme si teda príklad, ako vyzerá vytváranie intervalov. Ak máme záznam o narodení dieťaťa z 18.8.1860 a chceme spraviť odhad pre matku. Vieme že musela mať minimálne 15 rokov, keď sa dieťa narodilo, preto sa mohla narodiť najskôr pred 15 rokov. Zároveň

vieme, že maximálny vek, kedy sa mohla stať matkou bol 45 rokov, preto sa mohla narodiť najneskôr pred 45 rokmi. Matka mohla umrieť kedykoľvek po pôrode aj v daný deň. Ak počítame, že maximálny vek osoby mohol byť 100 rokov,tak ak mala matka dieťa v 15rokov, tak mohla žiť ešte 85 rokov. Výsledky nášho uvažovania sme spracovali do obrázku 3.11. Vznikol nám interval narodenia a interval úmrtia pre matku.



Obr. 3.11: Intervaly narodenia a úmrtia

Ak by sme našli napríklad zhodu v zázname o narodení matky, tak by sa dané intervaly upravili nasledovne. Interval narodenia by sme zredukovali už iba na jeden dátum v našom prípade 11.7.1841. Tento dátum sa nachádza v predošlom intervale narodenia matky. Následne skontrolujeme interval pre úmrtie. Vieme, že matka sa narodila v roku 1841 a predpokladáme, že sa dožila najviac 100 rokov. V roku 1945 by však mala už 104 rokov. Preto zmenšíme interval úmrtia. Upravené intervaly môžeme vidieť na obrázku 3.12.



Obr. 3.12: Aktualizované intervaly narodenia a úmrtia

3.3.2 Porovnávanie záznamov

V nasledujúcej časti si priblížime porovnávanie záznamov. Ako prvé si overíme, že osoba v čase vytvárania záznamu žila. Potom môžme prejsť k ďalšiemu kroku porovnávania. Tým je opätovne z dôvodu ušetrenia si detailnejšieho porovnávania jednoduchá kontrola základných údajov. Vezmeme si iba informácie z grafovej databázy, ktoré sme mali uložené v uzle a všetky mestá v ktorých osoba žila. Na základe nich porovnáme meno, priezvisko, povolanie a mesto. Všetky tieto údaje máme uložené aj v normalizovanej forme, preto budeme zisťovať, či sú totožné medzi porovnávanými osobami. Ak nenájdeme žiadnu zhodu, teda všetky hodnoty sú rozdielne, nebudeme porovnávať ďalej. V prípade, ak nájdeme aspoň nejakú zhodu prejdeme k detailnejšiemu porovnávaniu. Tento krok robíme z toho dôvodu, aby sme sa vyhli veľkému množstvu operácií nad grafovou databázou. Zmieňovanými podmienkami si zabezpečíme relevantnú skupinu, ktorú má zmysel detailnejšie porovnávať. Pri porovnávaní osôb sa snažíme porovnať všetky nám dostupné údaje aj v rámci rodinných príslušníkov.

Dáta, ktoré máme v normalizovanej forme porovnávame jednoduchým spôsobom, buď ide o zhodu alebo nie. Ak sa z nejakého dôvodu normalizovaná hodnota v databáze nenachádza porovnávame pomocou Levenshteinova vzdialenosť2.2.4. V prípade, ak ideme porovnávať ženské priezviská odstránime príponu -ová. Ak by boli priezviská krátke, tak by nám v prípade neodstránenia vyšla neprimerane veľká zhoda.

Pri porovnávaní miest máme k dispozícii okrem metódy klasického porovnávania a porovnávania na základe Levenshteinovej vzdialenosti aj možnosť využiť vzdialenosť miest.

Ak výsledkom porovnávacích metód nebude úplná zhoda, použijeme GPS súradnice miest. Súradnice nám poslúžia na vypočítanie vzdialenosti medzi mestami. Získame tak počet kilometrov medzi mestami. Následne pomocou exponenciálnej funkcie si určíme pravdepodobnosť, že ide o mestá blízko seba. Čím sa mestá nachádzajú bližšie, tým je pravdepodobnosť vyššia.

Ďalším typom porovnávania je porovnávanie dátumov. Z dátumu si vytvoríme sústavu čísel bez znakov. To znamená, že z dátumu 10-04-1865 vytvoríme 10041865. Takto upravené dátumy porovnáme rovnako, ako slová pomocou Levenshteinovej vzdialenosti. Pri dátumoch skontrolujeme, či náhodou nie je zamenený deň a mesiac. V prípade, ak porovnávame dátumy narodenia prepočítame si ich na vek. Následne si vypočítame vzdialenosť týchto údajov 2.2.4. Na konci si vyberieme hodnotu, ktorá vyšla ako najlepšia.

Jediným číselným údajom je číslo domu, to si porovnáme pomocou číselného porovnávania 2.2.4. Okrem toho ho porovnáme aj ako slovo pomocou Levenshteinovej vzdialenosti. Ako výsledok vyberáme najlepšiu hodnotu.

Slová ktorých normalizovanú formu nemáme napríklad ulica, viera, titul porovnávame pomocou Levenshteinovej vzdialenosti. Značky respektíve hodnoty, ktoré môžu nadobúdať iba hodnoty 1 a 0. Ktoré hovoria o tom, či daný výrok platí alebo nie. Porovnávame ich jednoducho, buď sú rovnaké alebo nie. Medzi značky patria premenné *vícerčata, otec_mrtev, nalezenec* a *manželské*.

Premenná	Typy porovnávania
Meno	Presná zhoda, Levenshteinova vzdialenosť
Priezvisko	Presná zhoda, Levenshteinova vzdialenosť
Povolanie	Presná zhoda, Levenshteinova vzdialenosť
Mesto	Presná zhoda, vzdialenosť miest
Ulica	Levenshteinova vzdialenosť
Popisné číslo	Levenshteinova vzdialenosť, porovnávanie čísel
Dátum narodenia	Levenshteinova vzdialenosť, kontrola dátumov, porovnanie veku
Dátum (zvyšné typy)	Levenshteinova vzdialenosť, kontrola dátumov
Titul	Levenshteinova vzdialenosť
Viera	Levenshteinova vzdialenosť
Značky	Presná zhoda

Tabuľka 3.5: Prehľad premenných a spôsobov ich porovnávania

V tabuľke 3.5 môžme vidieť prehľad premenných a to akými spôsobmi ich porovnávame. V prípade, ak je osoba z grafovej databázy už vytvorená z niekoľkých osôb, môže mať niektoré hodnoty viacnásobné. Napríklad môže mať viac zamestnaní. V tom prípade porovnávame všetky hodnoty a vyberieme tú s najvyššou hodnotou. Teda tú, kde je najlepšia zhoda.

Následne na základe toho akých predkov má osoba v zázname, začneme prehľadávať, či aj porovnávaná osoba v grafovej databáze nemá rovnakých predkov. Načítame teda z databázy ďalšie osoby a porovnávame ich rovnakým spôsobom. Snažíme sa pritom porovnávať, čo najviac do histórie, ako nám to daný záznam dovolí. Týmto spôsobom budeme vedieť s väčšou určitosťou prepájať ľudí.

Klasifikácia záznamov

Keď máme takto porovnané hodnoty, môžme pristúpiť ku klasifikácii. Na základe 2.2.5 budeme sčítavať všetky výsledky porovnávania. Tie budeme násobiť váhami. Aby sa všetky hodnoty držali v rozsahu 0 až 1 aj váhy budú v tomto rozsahu. Údaje o osobách, ktoré sa nemenia budú mať váhu 1. To je napríklad meno, priezvisko, dátum narodenia alebo úmrtia. Dáta, ktoré sa počas života môžu meniť, budú mať váhu 0.5. Osoba môže zmeniť počas života adresu alebo povolanie. Dáta, ktoré má väčšia osôb rovnakých a preto nemajú veľký vplyv zase nastavíme na 0.1. V tomto prípade ide hlavne o značky. Napríklad či otec bol pri narodení dieťaťa mŕtvy alebo vieru, keďže ju má väčšina ľudí rovnakú.

O každej osobe môžeme mať rôzne množstvo informácii. O krstencovi môžeme mať veľmi podrobné informácie s množstvom dát o jeho príbuzných. Na druhú stranu u pôrodnej baby môžeme mať iba informáciu o jej mene priezvisku a nič viac. Kvôli tomu musíme rozlišovať okrem dosiahnutej zhody aj relevantnosť záznamu. Preto okrem výsledku porovnávania budeme počítať okrem klasifikácie aj spoľahlivosť záznamu. Čím viac hodnôt bude vyplnených, tým bude záznam spoľahlivejší.

Aby sme výsledok porovnávania udržali v rozmedzí 0 a 1 budeme teda výslednú hodnotu porovnávania deliť maximálnou možnou dosiahnuteľnou pre daný záznam. Teda vždy, keď budeme porovnávať hodnoty, pripočítame si do maximálne dosiahnuteľného skóre aj hodnotu $1 \times w$. Teda hodnotu v prípadne úplnej zhody, krát váhu daného atribútu.

Výslednú hodnotu budeme porovnávať so zvolenými prahmi t_l a t_u. Hodnoty týchto prahov sa budú experimentálne overovať. Budeme sa snažiť zistiť, ako správne nastaviť prahy, aby sme dosiahli čo najlepšie výsledky klasifikácii.

3.3.3 Vyhodnotenie porovnávania

Na základe výsledku porovnávania sa rozhodneme, ako budeme ďalej postupovať. Ak nenájdeme žiadnu zhodu medzi osobou, ktorú porovnávame a všetkými osobami v grafovej databáze, vytvoríme nový uzol so všetkými informáciami o osobe.

V prípade nájdenia potencionálnej zhody vytvoríme nový uzol a prepojíme ho už s existujúcim. Pričom k prepojeniu uložíme výslednú hodnotu porovnávania. Potom budeme vedieť s ktorými osobami je teda osoba najpravdepodobnejšie zhodná. Vzťahy vrámci záznamu prepájame s novým uzlom nezávisle od starého.

Ešte pred samotným prehlásením, že osoby sú zhodné musíme skontrolovať, či je možné, že ide o danú osobu. Skontrolujeme, či sa nesnažíme priradiť druhé dieťa matke v období kratšom ako 9 mesiacov. Či sa dieťa nenarodilo otcovi neskôr ako 9 mesiacov po smrti. Ak nie je problém s týmito pravidlami, tak prehlásime osoby za zhodné. Uložíme do existujúceho uzlu údaje, ktoré ešte neobsahuje.

Keď nájdeme jednu zhodu, tak vieme, že môžeme nájsť aj ďalšie. Preto vrámci záznamu začneme porovnávať ďalšie osoby s tými, ktoré sú prepojené na nami nájdenú zhodu v grafovej databáze. Ak nájdeme zhodu v matke pri zázname o krste tak je pravdepodobné, že aj otec a ďalší členovia rodiny sa budú zhodovať. Nemusíme, preto prehľadávať zase celú databázu. Takto prepojíme časti rodokmeňov. V prípade, ak pôjde o kňaza, alebo pôrodnú babu, nebudeme prehľadávať ostatné osoby v zázname, pretože tieto osoby vystupujú v mnohých záznamoch a budú prepájať mnoho rodín v určitej oblasti. Avšak svedkovia, alebo krstní rodičia mohli byť členmi rodiny a preto skontrolujeme ich prepojenia.

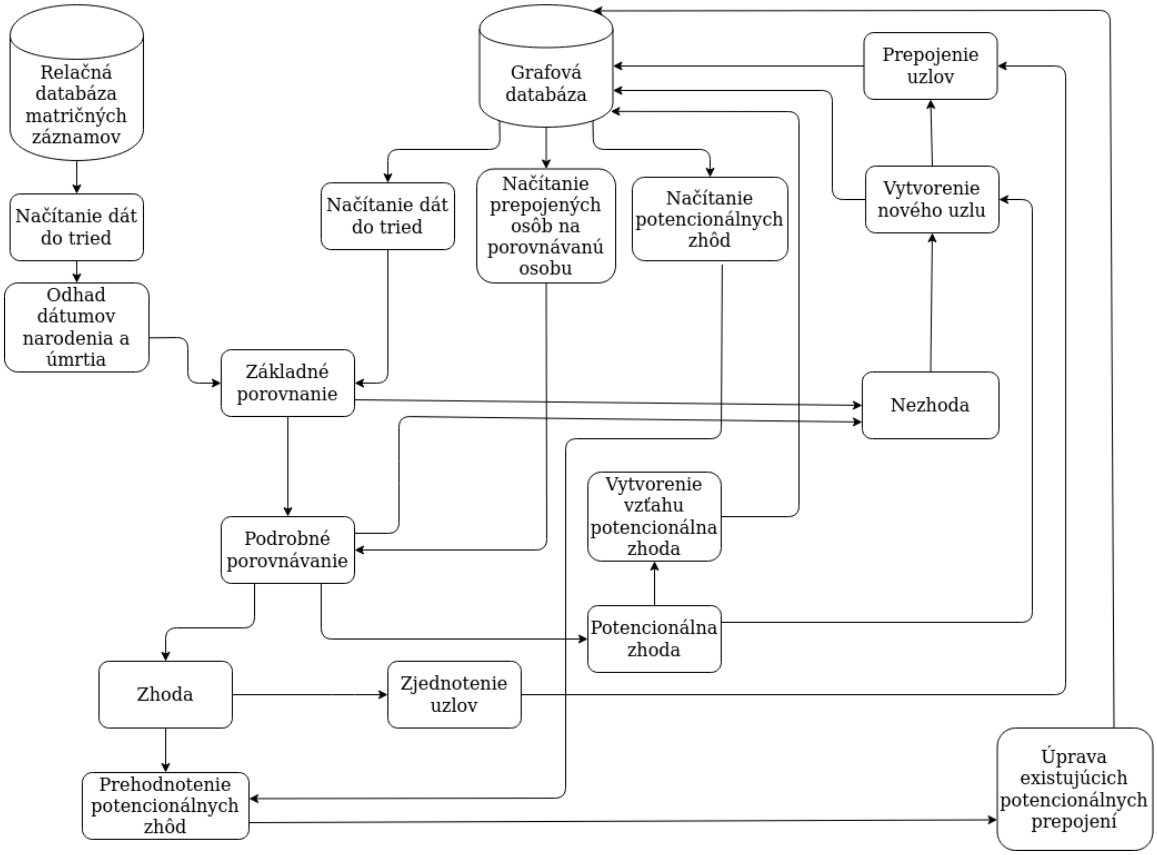
Záznam prepojíme s matrikou, buď už existujúcou, alebo vytvoríme novú. Pri prepájaní osôb s ich bydlisko kontrolujeme, či náhodou už daný uzol neexistuje. Túto kontrolu vykonávame na všetkých úrovniach adresy. Teda ak vieme, že neexistuje uzol z popisným číslom

13 s chýbajúcou ulicou, v meste Brno vytvoríme uzol číslo 13. Potom však kontrolujeme, či už existuje uzol Brno, aby sme nevytvárali duplicitné uzly s rovnakou informáciu.

Uzlom o osobách vytvoríme prepojenia medzi sebou na základe záznamu. Rovnako prepojíme aj záznam s osobami. Vzťahy, ktoré vytvoríme sme si už rozobrali pri návrhu grafovej databáze 3.2.1. Prepojenie osôb bude prebiehať teda na čiastočne už existujúcimi a čiastočne novými uzlami. Môžeme nájsť zhodu iba pre časť osôb zo záznamu. Môžu sa však vyskytnúť prípady, kedy sa už všetky osoby v databáze nachádzajú, ale aj také kde sa nenachádza žiadna osoba.

3.3.4 Výsledný návrh systému

Obrázok 3.13, sumarizuje celý opísaný návrh systému. Môžme vidieť ako postupujeme pri porovnávaní a zachádzame z výsledkami. Z obrázku je zreteľné, že budeme veľa pracovať s navrhnutou grafovou databázou.



Obr. 3.13: Podrobný návrh systému

Kapitola 4

Implementácia

Táto kapitola bude venovaná implementácii navrhnutého systému. Najskôr si rozoberieme použité technológie a knižnice, ktoré sme využili. Pre lepší prehľad si prejdeme všetky triedy, z ktorých sa výsledný systém skladá a ich základnú funkčnosť. Následne si prejdeme kľúčové časti programu. Rozoberieme si prácu s relačnou a grafovou databázou a porovnávanie záznamov. Poslednou časťou tejto kapitoly sú výsledné programy, vďaka ktorým vieme pristupovať k spracovaným dátam,

4.1 Použité technológie

Navrhnutý systém je implementovaný v jazyku Python 3. Medzi jeho výhody patrí dobre čitateľná syntax, či jednoduchosť. Poskytuje nám široké možnosti na spracovanie dát a prácu s databázami. Okrem dát z relačnej databázy máme k dispozícii aj iné zdroje dát. Súradnice GPS sa nachádzajú v samostatných JSON súboroch. Skratka JSON je odvodená z anglických slov JavaScript Object Notation teda JavaScriptový objektový zápis. Ide o štrukturovaný spôsob zápisu dát určený pre prenos dát. Testovacie dáta sú v CSV súboroch. Skratka CSV je odvodená od anglického Comma-separated values, čo predstavuje hodnoty oddelené čiarkami. Sú to jednoduché súbory, v ktorých sú dáta uložené vo forme čistého textu. Slúžia na ukladanie tabuľkových dát.

Mysql

Relačná databáza je typu MySQL ide o open source databázový server. Nachádzaná sa na školskom serveri perun. Na pripojenie do databázy sme využili ovládač mysql [1]. Na dotazovanie nad databázou sa využíva jazyk SQL.

Neo4j

Najznámejšou a najpoužívanejšou grafovou databázou je Neo4j. K prepojeniu s grafovou databázou využívame ovládač s príznačným názvom Neo4j Python [2]. Ide o oficiálne podporovaný driver pre pripojenie sa na Neo4j databázu[3]. Využíva k tomu binárny protokol. Ide o veľmi jednoduchý driver, s ktorým sa ľahko pracuje. Pomocou neho sa vieme dotazovať na databázu. Na prácu s Neo4j databázou sa využíva dotazovací jazyk Cypher. Príkazy v ňom si predstavíme vrámci implementácie.

[1] https://dev.mysql.com/doc/connector-python/en/
[2] https://neo4j.com/docs/api/python-driver/current/

Knižnice využité pri implementácii

Pre jazyk Python existuje veľké množstvo knižníc. Pre našu prácu boli využité nasledovné:

- Levenshtein[3] - výpočet Levenshteinovej vzdialenosti medzi slovami
- geopy[4] - výpočet vzdialnosti medzi GPS súradnicami
- numpy[5] - výpočet exponenciálnej funkcie
- date - práca s dátumami
- json - spracovanie súborov v json formáte
- pandas[6] - spracovanie testovacích cvs súborov
- time - meranie času výpočtu

4.2 Štruktúra programu

Predtým ako prejdeme k samotnému programu predstavíme si jeho časti. Program je rozdelený do niekoľkých logických celkov, aby každý spracovával určitú úlohu. V tabuľke 4.1 môžme vidieť všetky súbory a ich základnú funkčnosť.

4.3 Algoritmus

Prvý krok je pripojenie sa do databáz. K pripojeniu do relačnej Mysql databázy využívame knižnicu mysql. S databázou sme pracovali lokálne v počítači. K pripojeniu do grafovej neo4j databázy využívame knižnicu neo4j. Prihlasovacie údaje do databáz sa nachádzajú v konfiguračnom súbore. Pri testovaní sme používali verziu databázy, ktorá sa nachádza na serveri Perun. V prípade úspešného pripojenia, môžme začať s dotazmi nad databázami. Ak sa nám nepodarí pripojiť, program vypíše chybu.

4.3.1 Načítanie záznamu

Po pripojení je ďalším krokom získanie všetkých záznamov o narodení. Všetka práca s touto databázou sa vykonáva v triede *relation_database_birth.py*. Nespracované záznamy načítavame z relačnej databázy. Načítame a vytvoríme si objekty osôb na porovnávanie. Tieto osoby budeme porovnávať s tými, ktoré sú už uložené v grafovej databáze.

Načítanie z relačnej databázy

Program načíta všetky tabuľky *birth* 3.2, ktoré sme si rozobrali v sekcii o relačnej databáze 3.1. Každá táto tabuľka reprezentuje jeden záznam. Z informácii ktoré sú v nej uložené, si vytvoríme nový objekt *register*, do ktorého si uložíme dáta o zázname. Ďalším krokom je získanie informácii o matrike. Pomocou SELECT dotazov získam všetky údaje a uložím si ich do objektu *register*.

[3] https://pypi.org/project/python-Levenshtein/
[4] https://geopy.readthedocs.io/en/stable/
[5] https://numpy.org/
[6] https://pandas.pydata.org/

Triedy	Význam triedy
main.py	Základný súbor, spracováva pripojenia na databázu a riadi testovanie
create_database.py	Hlavný algoritmus celého programu riadi načítavanie záznamov, vytváranie grafovej databázy a spracováva výsledky porovnávania
relational_database_birth.py	Načítava dáta z relačnej databázy
graph_database.py	Načítava, aktualizuje, vyhľadáva, vytvára uzly a prepojenia v grafovej databáze
csv_source.py	Načítava dáta z CSV súborov
comparator.py	Porovnáva osoby a vyhodnocuje výsledky, obsahuje všetky metódy potrebné k porovnávaniu
record.py	Trieda spravujúca záznam, obsahuje vytváranie uzlu, aktualizovanie uzlu, výpis informácii
person.py	Trieda spravujúca osobu, vytváranie uzlu, aktualizovanie uzlu, výpis informácií
register.py	Trieda spravujúca matriky, vytváranie uzlu, aktualizovanie uzlu, výpis informácií
domicile.py	Trieda spravujúca adresy, prehľadávanie json súborov, výpis informácií
date.py	Trieda spravujúca dátumy
get_persons.py	Skript na základe mena a priezviská nájde všetky zodpovedajúce osoby uložené v grafovej databáze
get_all_records.py	Skript na základe id osoby nájde všetky záznamy v grafovej databáze, v ktorých sa osoba nachádza
get_family_tree.py	Skript na základe id osoby vypíše všetkých predkov a potomkov ktorý sa nachádzajú v grafovej databáze

Tabuľka 4.1: Prehľad tried a ich význam

Následne pomocou metódy *get_persons* získame všetky tabuľky *birthPerson*. Postupne cez ne prejdeme a pomocou ďalších dotazov vytvoríme objekt *person* so všetkými informáciami. V sekcii 3.1.2, je podrobne rozobraté, aké informácie vieme získať a kde sa nachádzajú. Všetky tieto údaje ukladáme do zoznamov. Zo získaných adries vytvárame samostatné objekty *Address* rovnako aj dátumy ukladáme do triedy *Date*. Podľa dátumu vzniknutia záznamu odhadneme dátum narodenia a úmrtia. Vytvoríme nové vlastnosti triedy person a to *birth_date_guess* a *dead_date_guess*. Tieto premenné sú typu *dictionary* a obsahujú atribúty dátum *from*, *to* a *correct*. Pričom hodnoty *from* a *to* sú typu *Date* a určujú nám interval. Pomocou príznaku *correct* si vieme overiť, či ide o presný dátum. Tieto odhady budú tým presnejšie, čím viac informácií budeme mať o osobe. Takto vytvoríme zoznam osôb na ďalšie spracovanie a uložíme si ho do triedy *Record*. Posledným krokom je doplnenie informácie o hlavnej osobe záznamu, ktorá je určená vzťahom *main*. V tabuľke *birth* sú o nej dostupné ešte doplňujúce informácie.

Vo výslednej triede *record*, sa nachádzajú všetky dostupné informácie o zázname, osobách a matrike.

Načítanie z grafovej databázy

Pri načítaní z grafovej databáze sa sústredíme na osoby a nie na záznamy. Takže načítavame všetky osoby a postupne ich spracovávame. Keďže porovnávame podľa pohlavia, načítame osoby na základe neho. Pomocou dotazu vieme získať všetky uzly daného pohlavia. Dotazy nad grafovou databázou vyzerajú nasledovne:

```
MATCH (person:Osoba)

WHERE 'U' IN person.pohlaví

RETURN person
```

V tomto prípade nájdeme všetky osoby, ktoré majú nedefinované pohlavie. Skratka U je odvodená z anglického slova undefined teda nedefinované. Dotaz nám vráti zoznam všetkých uzlov, pre ktoré platí takéto pravidlo. Uzly sú typu *Node*. Ide o typ preddefinovaný Neo4j databázou. Keď už máme takto načítané uzly, môžme prejsť na ich uloženiu do objektu *Person*. Tento krok je potrebný kvôli jednoduchšiemu prístupu k údajom. Keďže všetky údaje okrem adresy sú uložené už v danom uzle načítanie hodnôt je výrazne jednoduchšie. Stačí použiť funkciu get() nad uzlom, ktorý je spomínaného typu Node a získame hodnotu, ktorú zadáme do funkcie.

Z návrhu grafovej databáze 3, vieme že adresa sa neukladá priamo do uzlu, ale je určená vzťahom. Ak teda chceme vedieť, kde osoba bývala potrebujeme získať všetky jej prepojenia vzťahom *BÝVÁ*. Dotaz nad databázou vyzerá nasledovne.

```
MATCH (person:Osoba)-[relation:BÝVÁ]->(address)

WHERE ID(person) = person.id

RETURN address
```

Môžme vidieť, že hľadáme všetky uzly typu *Osoba* prepojené vzťahom *BÝVÁ* na nejaký ďalší nedefinovaný uzol. Hľadáme však konkrétny uzol s daným ID. Takto vieme získať uzol, na ktorý je osoba prepojená vzťahom, ktorý chceme. Keďže nevieme ako presnú informáciu dostaneme, či máme popisné číslo, ulicu alebo iba mesto musíme sa prípadne znova dotazovať. Teraz však už budeme hľadať prepojenia *JE_V* medzi časťami adresy. Prípadne ak má osoba viac adries načítame všetky. Údaje ktoré načítame postupne z uzlov uložíme do triedy *Domicile*.

Dotazy nad grafovou databázou sú ľahko čitateľné a veľmi intuitívne.

Načítanie GPS súradníc

Pre výpočet vzdialenosti miest je potrebné vedieť ich GPS súradnice. Tie sa nenachádzajú v relačnej databáze ale v JSON súboroch. Nenachádzajú sa v nich však GPS súradnice všetkých obcí. K dispozícii sme dostali dva súbory pre Juhomoravský a Moravsko-sliezsky kraj. Pomocou knižnice json prejdeme celé súbory a v prípade, ak tam nájdeme súradnice priradíme ich obci a uložíme do grafovej databázy. Funkciu *set_gps_coordinates*, ktorá ma na starosti nastavenie týchto súradníc voláme hneď po priradení mesta osobe.

4.3.2 Porovnávanie záznamu

Keď už máme načítané osoby, môžme prejsť k porovnávaniu záznamov. Keďže to prebieha na základe pohlavia načítame si všetkých mužov, ženy a neidentifikované osoby. Potom na základe toho s kým plánujeme porovnávať si vyberieme skupiny ľudí. Osoby, ktoré nemajú definované pohlavie porovnávame aj so ženami aj s mužmi.

Metóda *get_result_of_comparison*, nám vracia výsledok porovnávania. Postupne prechádzame všetky vybraté osoby z grafovej databázy a porovnávame ich. Prvým krokom porovnávania je už spomínaná kontrola, či daná osoba mohla v čase vytvorenia záznamu žiť. Porovnáme teda dátum vytvorenia záznamu s odhadom narodenia a úmrtia. V tomto prípade počítame s tým, že je záznam vytvorený v deň krstu. Ak osoba žila, pokračujeme ďalej v porovnávaní, ak nie tak pokračujeme na ďalšiu osobu.

Ďalším krokom je základné porovnanie. Skontrolujeme základné údaje ako sú meno, priezvisko, povolanie a mesto. Ak sú všetky tieto údaje odlišné, tak nepokračujeme ďalej v porovnávaní. Ak sa však aspoň jedna hodnota zhoduje môžme prejsť k detailnému porovnaniu ľudí.

Z triedy *comparator* zavoláme teda metódu *detailed_comparison_of_two_persons* a tá nám porovná všetky údaje, ktoré máme u osôb uložené. K výpočtu Levenshteinovej vzdialenosti sme využili knižnicu *Levenshtein*. Všetky výsledky porovnávaní normalizujeme do intervalu 0 až 1. Vytvoríme si slovník *comparison*, do ktorého uložíme najlepší výsledok porovnávania pre každý atribút. Takto vieme pristupovať pomocou kľúčového slova k výsledku porovnávania daného atribútu. V prípade, ak máme viac hodnôt porovnávame teda každú hodnotu s každou. Ak sme nemohli vykonať porovnávanie, z dôvodu chýbajúcej hodnoty výsledok je -1. V prípade, ak chýbala normalizovaná hodnota a existuje nenormalizovaná, tak počítame Levenshteinovu vzdialenosť.

Na základe role vieme, že môžme hľadať ďalšie zhody v grafovej databáze. V prípade dieťaťa na základe rodičov a starých rodičov. Ak ide o rodiča, tak na na základe jeho rodičov. Po porovnaní osoby vieme teda v niektorých prípadoch pokračovať v porovnávaní. Vyhľadáme si v rámci matričného záznamu rodiča osoby a rovnako aj v grafovej databáze skúsime nájsť rodiča porovnávanej osoby. Následne aj pre neho vytvoríme slovník *comparison*. Takto porovnávame všetky dostupné osoby na základe vzťahov *JE_OTEC* a *JE_MATKA*. Porovnávame až tak ďaleko vrámci predkov, ako nám to spracovávaný matričný záznam dovolí. Všetky tieto porovnávania si uložíme do zoznamu na výpočet výsledného porovnávania.

4.3.3 Klasifikácia záznamu

Po tom čo máme už všetky údaje porovnané je čas na spočítanie výsledku. Postupne prechádzame všetky údaje v slovníku *comparison* a hodnoty, ktoré sú -1 ignorujeme, pretože neboli vyplnené. Ostatné výsledky násobíme podľa atribútu s váhou a sčítavame hodnoty do celkového výsledku. Popri tom sčítavame aj potencionálne možný najlepší výsledok. Keď prejdeme všetky výsledky vydelíme hodnoty a výsledné skóre porovnáme s určenými prahmi.

4.3.4 Uloženie záznamu

Prvým krokom je vytvorenie samotného uzla *Záznam_o_křte*. Následne skontrolujeme či sa náhodou už nenachádza matrika, v ktorej je uložený záznam. Túto kontrolu vykonávame na základe ID z relačnej databázy. Ak sa matrika ešte v grafovej databáze nenachádza, tak

ju vytvoríme a prepojíme s novo vytvoreným záznamom. Ak sa už nachádza, vytvoríme iba prepojenie *JE_V*. Následne prejdeme na vytváranie osôb.

Z výsledkov teda už vieme či šlo o zhodu, potencionálnu zhodu, alebo nezhodu. Preto v triede *create_database.py* v metóde handle_result_comparation.py po získaní výsledku sa na základe neho rozhodne o ďalšom kroku.

V prípadoch, ak sme nenašli zhodu, alebo sme našli potencionálnu zhodu vytvoríme nový uzol. Z triedy *person.py* zavoláme metódu *create_node_person*, ktorá vytvorí dotaz *CREATE* nad grafovou databázou. Tento dotaz vytvorí nový uzol so všetkým údajmi o osobe. Údaje ukladá už do spomínaných listov. Výnimku tvoria odhady dátumov narodenia a úmrtia. Po vytvorení uzlu prichádza na rad vytvorenie adresy a jej prepojenie.

Pre vytvorenie adresy potrebujeme kontrolovať, či už nie je uložená v databáze. V prípade, ak už niekto býva na rovnakej adrese nechceme vytvárať zbytočne rovnaké uzly. Preto vždy pri vytváraní uzlu kontrolujeme, či už daný uzol neexistuje. Ak budú takto prepojené uzly, budeme napríklad vedieť o všetkých ľudoch, ktorý na danej adrese v priebehu času bývali. Pri kontrole existencie uzlov musíme opätovne postupovať po úrovniach. Teda ak máme popisné číslo kontrolujeme, či sa nenachádza kombinácia popisné číslo, ulica a mesto. Prípadne kontrolujeme kombináciu popisné číslo a mesto. Záleží samozrejme na údajoch, ktoré máme k dispozícii. Po vytvorení uzlov prepájame najdetailnejšiu informáciu vzťahom *BÝVA* na osobu. K vzťahu pripojíme ešte údaj o dátume, kedy tam daná osoba žila. Zvyšok adresy hierarchicky k všeobecnejšej informácii prepojíme vzťahom *JE_V*.

Ak sme našli potencionálnu zhodu, vytvoríme si nový uzol, ako v predchádzajúcom prípade, a prepojíme ho s existujúcou osobou vzťahom *POTENCIONALNI_ZHODA*.

Pri nájdení zhody si nájdeme zhodujúcu sa osobu v grafovej databáze a z triedy *person.py* zavoláme metódu *update_node_person*. Tá pomocou príkazu *SET* aktualizuje iba hodnoty, ktoré sa nezhodujú s už uloženými. V prípade ak sa odlišujú niektoré údaje, tak si ich pridáme do zoznamu napríklad osoba môže mať viac povolaní. Takto nestratíme žiadne informácie a získame obšírnejší záznam. Okrem údajov o osobe ukladáme aj všetky ID z relačnej databázy.

Keď rozšírime uzol o ďalšie hodnoty, môže nám to ovplyvniť potencionálne zhody. Niektorá zo zhôd sa môže stať zhodou prípadne nezhodou. Preto vždy po nájdení zhody, prepočítame všetky potencionálne zhody. V prípade zmeny výsledku upravíme buď iba výsledok porovnávania, alebo ak nám výsledok prelomil jeden z prahov zmeníme prepojenia. Ak nám vyšlo že uzly už viac nie sú zhodné vymažeme prepojenie. V prípade ak výsledok značí, že ide o zhodu zjednotíme uzly. Potom opätovne prepočítame všetky potencionálne zhody či nám táto zhoda neovplyvnila ďalšie potencionálne prepojenia.

Keď už máme všetky uzly vytvorené, môžeme prejsť k ich prepájaniu. V metóde *create_connection_with_person_in_birth_record* budeme iterovať cez všetky osoby a vytvárať vzťahy. Na základe role v relačnej databáze začneme prepájať uzly osôb so záznamom. V tabuľke 3.1 sme si vymenovali všetky vzťahy, ktoré môžu vzniknúť. K vzťahom pripájame aj dátum záznamu. Popri prepájaní záznamu s osobou prepájame aj osoby medzi sebou. Podľa tabuľky 3.4 vytváram vzťahy medzi osobami.

Po vytvorení vzťahov máme spracovaný celý záznam a môžeme prejsť k načítaniu ďalšieho záznamu z relačnej databáze. Takto načítavame záznamy, kým nespracujeme celú databázu.

4.4 Výsledné programy

Po spustení navrhnutého algoritmu sa v grafovej databáze vytvoria časti rodokmeňov. Na základe porovnávania sa zjednotia niektoré osoby z relačnej databáze. Keď robíme genealogický výskum, tak nás zaujíma samozrejme rodokmeň. Okrem toho nás zaujíma v ktorých záznamoch sa osoba nachádza a aké role zastupuje. Pre tieto dotazy sme vytvorili skripty. Príklady použitia týchto skriptov sa nachádzajú v prílohe C.

4.4.1 Výpis osôb s daným menom

V prípade ak viete meno a priezvisko osoby, ktorú hľadáte zadáte tieto údaje ako parametre skriptu *get_person.py*. Tento program bude vyhľadávať vo vytvorenej grafovej databáze. Vypíše všetky osoby so zadaným menom. Zobrazí všetky dostupné informácie, ktoré sú o nich uložené. Vrámci týchto údajov vypíše aj ich ID. Na základe tohto ID potom pracujú nasledujúce dva skripty.

Ak by sme hľadali napríklad osobu s menom Adam Dostál dotaz by vyzeral nasledovne:

```
MATCH (person:Osoba)
WHERE (Adam IN person.meno OR Adam IN person.normalizovne_meno) AND
        Dostál IN person.příjmení OR Dostál IN person.normalizovne_příjmení)
RETURN person
```

Prehľadávame teda všetky osoby, ktoré majú zadané meno buď v pôvodnej podobe prípadne v normalizovanej.

4.4.2 Výpis všetkých záznamov

Skript *get_all_record.py* získa všetky záznamy v ktorých sa daná osoba nachádza. Vypíše všetky informácie o danej osobe, dátum kedy vznikol záznam a rolu ktorú osoba zastávala v zázname. K tomu však potrebujeme vedieť ID osoby. Bez neho by sme nevedeli určiť o ktorú osobu ide. Pomocou neho vieme presne určiť osobu o ktorú máme záujem, keďže ide o jedinečný identifikátor osoby. V databáze môže byť viac ľudí s rovnakými menami napríklad otec a syn. Preto je potrebné, najskôr na základe výpisu osôb nájsť osobu o ktorú máme záujem a až následne môžme použiť tento skript.

V tomto prípade ide o jednoduchý dotaz, ak by sme vyhľadávali osobu s ID 5 vyzeral by takto:

```
MATCH (person:Osoba)-[relation]->(record:Záznam_o_křte)
WHERE 5 IN person.id_relačná_databáza
RETURN record, relation
```

Vyhľadáme všetky záznamy prepojené na osobu s daným ID. Tento dotaz vykonáme pre všetky typy záznamov, príklad je pre záznam o krste. Pričom vrátime aj prepojenie týchto uzlov. Z ktorého potom budeme vedieť zistiť dátum záznamu a rolu, ktorú osoba zastávala.

4.4.3 Výpis príbuzných

Rovnako ako v predošlom prípade potrebujeme vedieť ID osoby. Tento skript hľadá všetky prepojenia osoby*JE_MATKA, JE_OTEC* a *JSOU_MANŽELÉ*. Najskôr vyhľadá manžela osoby. Pričom nesmieme zabudnúť, že osoba mohla byť za život vydaná aj viackrát. Následne vyhľadá všetky deti osoby, pričom ak nájde zhodu rekurzívne zavolá tú istú funkciu a hľadá deti nájdeného dieťaťa. Toto vykoná pre všetkých nájdených potomkov. Vytvorí teda rozrod pre danú osobu. Následne hľadá rovnaké vzťahy ale v opačnom smere. Hľadá všetkých predkov, čiže vytvára vývod. V prípade, ak nájde otca, alebo matku osoby opäť rekurzívne hľadá ich rodičov. Takto vieme nájsť všetkých príbuzných danej osoby. O každej nájdenej osobe vypíše prístupné údaje.

Pri hľadaní všetkých príbuzných používam 3 typy dotazov. Ak by sme vyhľadávali rodičov pre osobu s ID 15 vyzeral by takto:

```
MATCH (person:Osoba)-[relation]->(person1:Osoba)
WHERE 15 IN person1.id_relačná_databáza AND
        (type(relation)= 'JE_MATKA' OR type(relation)= 'JE_OTEC' )
RETURN person, relation
```

Nájdeme tak všetky osoby, ktoré sú prepojené vzťahom *JE_MATKA, JE_OTEC* na osobu, ktorú hľadáme. Naopak, ak by sme hľadali deti vyhľadávanej osoby, hľadali by sme rovnaké vzťahy, ale v opačnom smere. Teda z hľadanej osoby do ďalších uzlov. Takto dostaneme všetkých predkov, alebo potomkov.

Posledný dotaz nám vyhľadá manžela hľadanej osoby, ktorý vyzerá nasledovne:

```
MATCH (person1:Osoba)-[relation:JSOU_MANŽELÉ]->(person:Osoba)
WHERE 15 IN osoba.id_relačná_databáza
RETURN person, relation
```

Hľadáme teda všetky prepojenia *JSOU_MANŽELÉ*.

Kapitola 5

Testovanie

Posledným krokom je overenie správneho fungovania systému. Databáza, ktorá bola k dispozícii nemala vhodne spracované dáta na testovanie. Medzi uloženými údajmi chýbali dáta o tom, ako by mali byť správne prepojené osoby. Jedná sa o údaj, ktorý by nám jednoznačne potvrdil, ktoré osoby sú naozaj totožné a ktoré nie. Testovanie preto prebehlo nad CSV súbormi, v ktorých sú tieto informácie dostupné. Na základe nich potom vieme skontrolovať, či sme správne určili zhodu alebo nie.

5.1 Dátové zdroje

Na testovanie boli vytvorené súbory údajov zo záznamov o narodení pre jednu dedinu. V genealogickom softvéri existuje manuálne prepojených 1961 záznamov. Tie boli exportované do CSV súborov. Každej osobe je priradené ID. Vďaka nemu budeme kontrolovať či ide o zhodu alebo nie. Tieto záznamy sú z rokov 1607 až 1899.

V dátach sa nenachádzajú normalizované údaje. Program je však navrhnutý tak, že s nimi počíta. Avšak tým že tieto údaje boli umelo vytvorené stratili sa rozdiely, ktoré sa nachádzajú v databáze. Preto sú krstné mená, priezviská či povolanie implicitne upravené. Mesto síce nie je upravené ale tým, že sa jedná o osoby z jedného miesta nie je to potrebné. Predpokladáme, že výsledky budú veľmi podobné, ako by boli nad normalizovanými hodnotami. Keďže tento krok je v databáze spravený.

Keďže sú dáta vygenerované z manuálne prepojených dát, sú príliš obsiahle na rozdiel od toho ako vyzerajú v realite. Poznáme všetkých 4 prarodičov, čo však neodpovedá reálnym záznamov. Vytvorené sú preto dve skupiny dát. Prvé sa snažia napodobňovať matričné záznamy. Predkovia, ktorí neboli v matričnom zázname sú vymazaní. Takto nám vznikne zdroj dát viac zodpovedajúci realite. Druhá množina údajov sú iba tie záznamy, v ktorých sú všetci starí rodičia. To nám z pôvodných 1961 záznamov zmenšilo počet na 1097.

Nad týmito druhmi dát sú vytvorené ďalšie 4 typy dátových zdrojov. Na základe toho, koľko informácii sa ponechalo sa dáta rozdelil nasledovne:

1. Rodičia dieťaťa a všetci starý rodičia
2. Rodičia dieťaťa a otec matky
3. Rodičia dieťaťa a priezvisko matky za slobodna
4. Rodičia dieťaťa

	Otec	Matka	Otcov otec	Otcova matka	Matkin otec	Matkina matka
Starý rodičia	1-1	1-1	1-1	1-1	1-1	1-1
Otec matky	1-1	1-1	0-0	0-0	0-0	0-0
Priezvisko matky	1-1	1-1	0-0	0-0	1-1	0-0
Otec a matka	1-1	1-0	0-0	0-0	0-0	0-0

Tabuľka 5.1: Údaje ktoré obsahujú testovacie dáta

V tabuľke 5.1 môžeme vidieť prehľad, ktoré dáta sa nachádzajú v ktorom zdroji dát.

Prvá 1 alebo 0 značí krstné meno a druhé číslo značí priezvisko. Pričom 1 znamená, že danú hodnotu poznáme. Naopak 0 značí, že hodnota je neznáma.

Tieto dátové zdroje sú však o čosi chudobnejšie, čo sa týka informácii nachádzajúcich sa o jednej osobe. Pre jednu osobu môžeme mať nasledovné informácie:

- ID osoby
- Meno
- Priezvisko
- Povolanie
- Obec
- Ulice
- Popisné číslo
- Vierovyznanie
- Dátum narodenia

Vďaka ID osoby vieme kontrolovať zhodu. Pri krstencovi sú okrem toho ešte údaje o pohlaví a o tom či ide o viacerčatá. Pri ostatných osobách si odvádzame pohlavie od role.

Dátové zdroje boli poskytnuté od vedúceho tejto práce Ing. Jaroslav Rozman Ph.D., ktorý s nimi pracoval v článku [17].

5.2 Priebeh testovania

Ako prvé bolo potrebné vytvorenie ešte jednej triedy, ktorá načíta záznamy z CSV súborov. Využili sme k tomu knižnicu pandas, ktorá výrazne zjednodušuje prácu s CSV súbormi. Pomocou súboru *cvs_source.py* sme rovnako ako pri relačnej databáze po jednom načítavali matričné záznamy. Následne sme záznam rozobrali na všetky osoby a prešli k porovnávaniu. To prebieha na rovnakých metódach ako pri relačnej databáze. Zásadná zmena je však, že porovnávame nenormalizované hodnoty. Preto namiesto presnej zhody, porovnávame všetky reťazce Levenshteinovou vzdialenosťou. Ďalšou zmenou je samotná kontrola, či sme správne určili zhodu. Počítame teda hodnoty TP, FP, TN a FN. Okrem týchto hodnôt klasifikujeme porovnávanie aj do potencionálnej roviny. Takže počítame aj koľko hodnôt sa nám podarilo klasifikovať aj do tejto skupiny. V ideálnom prípade by hranice potencionálnej zhody boli nastavené tak, že všetky osoby by sme klasifikovali správne alebo nesprávne. Zvyšné by boli na rozmedzí teda zaradené do potencionálnych zhôd.

Program na testovanie spustíme jednoduchým prepínačom -test. Následne sa spustí program nad CSV súbormi. Automaticky sa púšťa testovanie nad všetkými súbormi. Výsledky testovania sa ukladajú do textového súboru. Pretože beh programu je časovo náročný, vypisuje sa počet spracovaných záznamov. Keď sa spracuje záznam vypíše výsledné štatistiky. Testovanie prebiehalo na serveri perun.

5.3 Výsledky testovania

Testovanie vhodných prahov

Odhady prahov som najskôr testovala na jednom súbore. Vybrala som si súbor Otec matky. Keďže ženy boli často definované otcom môžme povedať, že ide o záznamy pomerme zodpovedajúce realite. Okrem toho som vybrala sadu testov, ktorá zodpovedá reálnym záznamom, aby som mala výsledky viac podobné realite. Tento súbor obsahuje 2349 osôb.

Nastavené prahy som postupne zvyšovala o hodnotu buď o 0,2 alebo o 0,3. Pričom počiatočné prahy boli $t_u = 0{,}85$ a $t_l = 0{,}65$. Teda prahy s predchádzajúceho testovania. Maximálne prahy som nastavila na $t_u = 0{,}95$ a $t_l = 0{,}75$. So zvyšujúcou sa hodnotou t_u by sme dosahovali lepšie výsledky. Avšak potom by sme hľadali by už iba priamu zhodu. Prišli by sme tak o možnosť, že údaje o ľudoch sú mierne rozdielne ale stále ide o tú istú osobu. To by mohlo viesť k zavádzajúcim výsledkom.

	Recall	Precision	F-measure	Čas porovnávania [s]
$t_u = 0.85$ $t_l = 0.65$	0.99	0.53	0.69	628
$t_u = 0.87$ $t_l = 0.67$	0.98	0.52	0.68	628
$t_u = 0.90$ $t_l = 0.70$	0.97	0.53	0.69	1002
$t_u = 0.92$ $t_l = 0.72$	0.97	0.53	0.70	1122
$t_u = 0.95$ $t_l = 0.75$	0.98	0.55	0.68	1002

Tabuľka 5.2: Zdroje dát s príbuznými zodpovedajúcimi matričným záznamom a otcom matky

Výsledky testovania môžme vidieť v tabuľke 5.2. Vyplýva z nich, že čím je vyšší je prah t_u, ktorý určuje presnú zhodu, tým máme lepšie výsledky. Teda čím je prísnejšie pravidlo výberu hodnoty, tým je pochopiteľne menej osôb zle klasifikovaných ako zhoda.

Naopak môže sa zdať, že hodnota t_l nemá žiaden vplyv na hodnotu *Recall*. Je to z toho dôvodu, že množstvo osôb je vyradených z klasifikácie, ešte pred samotným porovnávaním s touto hodnotou. Pri prvotnej kontrole sa osoby vyhodnotia ako nezhodné. Preto parameter t_l nemá až taký viditeľný vplyv, pri tak malých zmenách. Vďaka tomu máme však menej potencionálny zhôd, čo je pre nás výhodné, pretože ich nemusíme znovu prehodnocovať.

V tabuľke 5.3 vidíme vplyv prahu t_l na výpočet, je viditeľný pri množstve potencionálnych zhôd.

	Zhoda definovaná ako potencionálna zhoda	Nezhoda definovaná ako potencionálna zhoda
$t_u = 0.85$ $t_l = 0.65$	17	1289
$t_u = 0.87$ $t_l = 0.67$	19	1473
$t_u = 0.90$ $t_l = 0.70$	22	1835
$t_u = 0.92$ $t_l = 0.72$	21	2031
$t_u = 0.95$ $t_l = 0.75$	21	1804

Tabuľka 5.3: Zdroje dát s príbuznými zodpovedajúcimi matričným záznamom a otcom matky

Vidíme, že väčšina potencionálnych zhôd má byť definovaná negatívne. Pozitívne však je, že celkového počtu klasifikácii 16932 je definovaných hodnôt pomerne málo.

Výsledky na vypočítaných prahoch testovania

Ďalšie testovanie prebehlo nad hodnotami odvodenými z predchádzajúceho testovania. Prebehlo nad nasledujúcimi hodnotami s nastavenými parametrami nasledovne:

- $t_u = 0.95$
- $t_l = 0.70$

	Recall	Precision	F-measure	Počet osôb	Čas porovnávania [s]
Starý rodičia	0.93	0.72	0.81	7679	4369
Otec matky	0.88	0.58	0.70	4388	3409
Priezvisko matky	0.89	0.61	0.73	3291	1842
Otec a matka	0.98	0.52	0.68	3291	3000

Tabuľka 5.4: Zdroje dát so všetkými príbuznými

Jedným z problémov pri testovaní, bol veľmi vysoký čas porovnávania jedného súboru. Z tabuľky 5.4 môžeme vidieť, že čas porovnávania jedného súboru sa pohybuje okolo hodiny. Pri reálnych dátach nad databázou, ktorá bude mať mnohonásobne viac záznamov, to môže byť značný problém. Jedným z dôvodov nárastu času je porovnávanie potencionálnych zhôd, to výrazne navyšuje čas porovnávania. Pretože potrebujeme porovnať opätovne všetky prvky pričom to nemusí mať výrazný vplyv na výsledok. Musíme sa preto snažiť, aby sme nemali zbytočne veľa potencionálnych zhôd. Ďalším problémom môže byť veľké množstvo detailných porovnávaní. Aj keď sa snažíme vyhnúť množstvu dotazov nad databázou, je porovnávanie časove náročne.

Môžme vidieť, že sa nám podarilo pomerne správne zaradiť hranicu pre negatívne výsledky. Hodnota *Recall* sa vo všetkých prípadoch pohybuje okolo 90%. To znamená, že sme minimum prvkov klasifikovali ako FN. Naopak *Precision* nám vraví, že hodnotu pre zhodu sa nám nepodarilo nastaviť až tak presne. V prípadoch, kde je málo informácii o zázname, máme veľa osôb chybne zadefinovaných ako zhodných.

	Recall	Precision	F-measure	Počet osôb	Čas porovnávania [s]
Starý rodičia	0.95	0.73	0.83	4434	1878
Otec matky	0.84	0.57	0.68	2349	722
Priezvisko matky	1.0	0.57	0.72	694	318
Otec a matka	0.97	0.64	0.77	1425	1189

Tabuľka 5.5: Zdroje dát s príbuznými zodpovedajúcimi matričným záznamom

V prípade, kde dáta zodpovedá matričným záznamom 5.5, vychádzajú výsledky podobne ako v predchádzajúcom prípade. Čo môžme považovať za pozitívne, keďže aj napriek chýbajúcim dátam, sme schopný pomerne dobre osoby klasifikovať.

Jedným z problémov s presnosťou môže byť, spájanie uzlov pri zhode. Následne po spojení uzlov sa nám nakopírujú dáta do jedného uzlu. Keď sa nám takto podarí zle priradiť osoby, nakopí sa nám množstvo chybných dát. Avšak vďaka rozširovaniu uzlov, vieme získavať nové dáta a vzťahy pri osobách.

Z testovaní nad dodanou sadou dát nám vyplýva, že pri dostatku dát systém pracuje pomerne presne. Darí sa mu pomerne dobre rozdeľovať dáta správne do skupín. Najväčší problém je s určovaním zhodných osôb. Množstvo osôb je zle označených za zhodných aj keď nie sú. Táto situácia nastáva najčastejšie pri nedostatočných dátach. V tomto prípade aj málo zhodných údajov s vysokou váhou nám dokáže vytvoriť vysokú zhodu.

Na reálnych dátach v databáze je však predpoklad, že by testovanie dopadlo o čosi lepšie. Jedným z predpokladov pre toto tvrdenie je ten, že v databáze máme normalizované hodnoty. Mali by sme presné zhody pri kľúčových parametroch. Boli by očistené od chybných dát a boli by presne zadané. Napríklad pri povolaní v CVS súbore sa vyskytuje hodnota: *tkadlec (1770), Richter und ... (1800)*. V relačnej databáze, sa však už takéto dáta nenachádzajú.

Na druhú stranu prahy t_u a t_l sú nastavené pre testovacie dáta. Pravdepodobné je, že v prípade ak by sme mali možnosť testovať relačnú databázu vyšli by nám rozdielne výsledky. Prahy by sme preto museli upravovať pre danú databázu. Rovnako aj váhy jednotlivých hodnôt sú nastavené podľa dostupných CSV súborov.

5.3.1 Ďalší vývoj

Odovzdaním diplomovej práce vývoj programu nekončí. Nasledujúcim krokom bude nahranie programu na školský server perun a jeho fungovanie v reálnom čase. Program sa bude pravidelne spúšťať nad databázou a kontrolovať, či nepribudli na server nové dáta. Tie, ktoré pribudli spracuje a následne označí, ako už spracované záznamy. Momentálne je spracovaná len jedna databáza krstov. Keď pribudnú ďalšie databázy, bude potrebné spracovať aj tie. Na týchto krokoch v práci budem ďalej pokračovať v spolupráci s mojím vedúcim práce.

Okrem toho sú tu ďalšie spôsoby, ako je možné vylepšiť prípadne obohatiť tento program. Bolo by zaujímave implementovať ďalšie spôsoby porovnávaní dát a testovať, ktorý ma najlepšiu úspešnosť. Okrem toho váhy atribútov sú nastavené na pevné atribúty, čo je však zavádzajúce. Nezvyčajné údaje by mali mať spravidla väčšiu pravdepodobnosť ako bežne používané. Okrem toho existujú aj ďalšie metódy klasifikácie, ktoré je možné otestovať. Najväčšou výzvou by však určite bol pokus o využitie neurónových sietí.

Ďalšou možnosťou rozšírenia, by mohlo byť pridanie grafického rozhrania. Informácie z výsledných skriptov by sa teda užívateľovi graficky zobrazili. Čo by bolo určite prínosom, hlavne čo sa týka výpisu rodokmeňu. Ktorý si automaticky podvedome vizualizujeme.

Kapitola 6

Záver

Cieľom diplomovej práce bolo navrhnúť, implementovať a otestovať systém pre generovanie rodokmeňov. V prvej časti práce podrobne rozoberám obor genealógie. Obsah a históriu matričných záznamov, s ktorými budem neskôr pracovať. Následne definujem postupne všetky kroky, ako sa bude systém vytvárať.

Cieľ práce sa podarilo naplniť a vytvoril sa funkčný systém na prepájanie záznamov. Systém spracováva dáta z relačnej databázy. Okrem toho spracováva aj testovacie CSV súbory. Načítané nespracované záznamy porovnáva s už existujúcimi v grafovej databáze. Následne na základe výsledku porovnávania vytvára uzly a prepája ich s už existujúcimi uzlami, alebo vytvára samostatnú časť rodokmeňu.

Výsledky testovania môžeme zhodnotiť pozitívne. Najlepšie výsledky program dosiahol pri obsiahlych záznamoch. Podarilo sa mu dosiahnuť presnosť cez 80%. V prípade záznamov s malým množstvom informácii, nám táto pravdepodobnosť klesá. Avšak je to logický dôsledok nedostatku údajov.

Avšak aby sme mali plnohodnotný systém na vytváranie rodokmeňov, potrebovali by sme okrem matričných záznamoch o krstoch aj záznamy o úmrtí a o manželstve. Tie by nám doplnili chýbajúce informácie a upresnili porovnávanie. Vedeli by sme potom lepšie zhodnotiť výsledky celého systému.

Nasledujúci vývoj bude určite zameraný na spracovanie ďalších typov dát. Na základe matrík zosnulých a oddaných, by mali byť opätovne vykonané testy pre správnosť systému. Okrem toho by bolo zaujímavé, porovnať aj ďalšie metódy prepájania záznamov s už existujúcim systémom. Vytvorené rodokmene v grafovej databáze nám ponúkajú veľa možností. Máme neobmedzené alternatívy nadväzovania ľudí na seba. Vieme ľahko zisťovať mnoho informácii, ktoré by sme v relačnej databáze zisťovali veľmi komplikovane. Zaujímavé by bolo teda využiť tieto možnosti, na získavanie rôznych informácii o osobách či miestach.

Pre mňa je najväčším prínosom tejto práce získanie vedomostí o používaní grafovej databázy. Okrem toho som výrazne rozšírila svoje vedomosti z oboru genealógie a z oblasti prepájania záznamov.

Literatúra

[1] GEDCOM XML. dec 2001, [Online; accessed 5.1.2020].
URL https://web.archive.org/web/20061116032407/http://www.familysearch.org/GEDCOM/GedXML60.pdf

[2] Matriky na internetu. 2015, [Online; accessed 5.1.2020].
URL http://www.genealogie.cz/aktivity/digitalizace/

[3] Using Neo4j from Python. 2020, [Online; accessed 26.5.2020].
URL https://neo4j.com/developer/python/

[4] Baženov, V.: Za bohatstvím našich předků. 2019, [Online; accessed 5.1.2020].
URL https://www.hledanipredku.cz/gruntovni-pozemkove-knihy/

[5] Christen, P.: *Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Berlin: Springer, 2012.

[6] David, J.: *Faminiae genealogia. Po stopách našich předků*.

[7] Dintelman, S.; Maness, T.: Reconstituting the Population of a Small European Town Using Probabilistic Record Linking: A Case Study. 07 2010.

[8] Doležal, M. N.: Ancetry tvorba rodokmenu. 2009, [Online; accessed 5.1.2020].
URL https://ancestry.nethar.com/index.php

[9] Fellegi, I. P.; Sunter, A. B.: A theory for record linkage. *Journal of the American Statistical Association*, ročník 64, č. 328, 1969: s. 1183–1210.

[10] Hříbek, D.; Rozman, J.: Poloautomatická normalizace slov z matričních záznamů. 2019.

[11] Lednická, B.: Rodopisné stránky.
http://rodokmen.nase-koreny.cz/matriky/obsah_matrik.htm, [Online; accessed 20.11.2019].

[12] Lednická, B.: *Sestavte si rodokmen pátráme po svých předcích*. Grada, 2012, ISBN 978-80-247-4069-0.

[13] Levenshtein, V. I.: Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, ročník 10, 1966, s. 707–710.

[14] Marek: GEDCOM. http://www.geni.sk/gedcom/, 2010, [Online; accessed 5.12.2019].

[15] Meyer, D.: MyHeritage automates record-matching as genealogy wars heat up. 2012, [Online; accessed 5.1.2020].
URL https://gigaom.com/2012/09/19/myheritage-automates-record-matching-as-genealogy-wars-heat-up/

[16] Peterka, J.: *Cesta k rodinným kořenům aneb Praktická příručka občanské genealogie*. Libri, 2006, ISBN 80-7277-307-0.

[17] Rozman, J.; Zbořil, F.: Persons Linking in Baptism Records. In *Workshop PAOS2018 and PASSCR2018 of JIST2018 conference*, november 2018, s. 43–54.

[18] Serva, M.; Petroni, F.: Indo-European languages tree by Levenshtein distance. *EPL (Europhysics Letters)*, ročník 81, č. 6, feb 2008: str. 68005, doi:10.1209/0295-5075/81/68005.
URL https://doi.org/10.1209%2F0295-5075%2F81%2F68005

[19] Sperát, P. I.: Rodokmeny - Rodokmeny a nakladatelství. https://www.sperat.cz/rodokmeny/, 2019, [Online; accessed 20.11.2019].

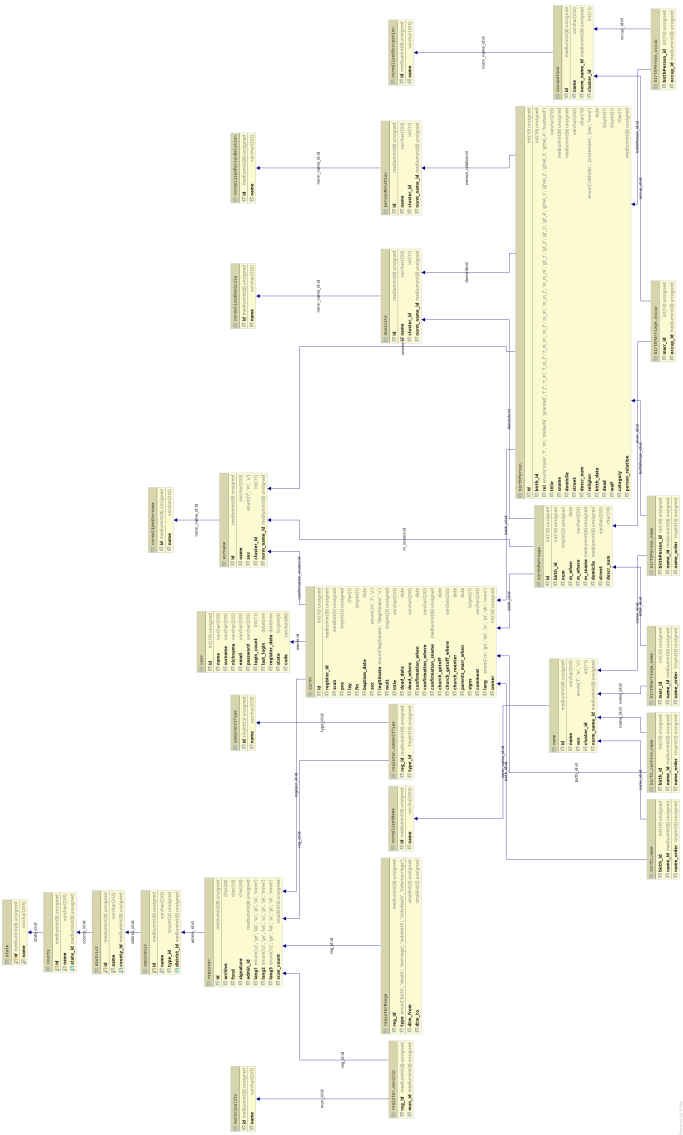
Príloha A

Obsah priloženého média

- Zdrojový kód programu
- Relačná databáza
- Testovacie CSV súbory
- Json zdrojové súbory
- Zdrojové súbory pre vytvorenie textu technickej správy
- Technická správa

Príloha B

ER diagram relačnej databázy



Obr. B.1: Návrh prepojovania záznamov

Príloha C

Príklad použitia programov

V nasledujúcej časti môžte vidieť výpisy výsledných programov, ktoré boli rozobraté v kapitole 4.4.

$ get_person.py -name Anton -surname Dostál

Hľadaná osoba Anton Dostál
.....
1. nalezena osoba

ID = 5
Jméno = Anton, Antonn - Antonín
Přímení = Dostal - Dostál
Pohlaví = U
Povolání = lokální kurát - Lokální kaplan

$ get_all_records.py -id 5

ID = 5
Jméno = Anton Antonn - Antonín
Přímení = Dostal - Dostál
Pohlaví = U
Povolání = lokální kurát - Lokální kaplan

Nalezena v následujících záznamech:
.....
1. záznam
Rola osoby v zázname = KRSTITEL
Dátum záznamu = 1860-02-09
Typ matričního záznamu = Záznam_o_křte
Sken = 10
Strana = None
Pozice = 1
Jazyk záznamu = GE
.....

2. záznam
Rola osoby v zázname = KRSTITEL
Dátum záznamu = 1860-03-15
Typ matričního záznamu = Záznam_o_křte
Sken = 10
Strana = None
Pozice = 2
Jazyk záznamu = GE
.....
3. záznam
Rola osoby v zázname = KRSTITEL
Dátum záznamu = 1860-03-19
Typ matričního záznamu = Záznam_o_křte
Sken = 10
Strana = None
Pozice = 3
Jazyk záznamu = GE
.....
4. záznam
Rola osoby v zázname = KRSTITEL
Dátum záznamu = 1860-04-07
Typ matričního záznamu = Záznam_o_křte
Sken = 10
Strana = None
Pozice = 4
Jazyk záznamu = GE
.....
5. záznam
Rola osoby v zázname = KRSTITEL
Dátum záznamu = 1860-04-24
Typ matričního záznamu = Záznam_o_křte
Sken = 10
Strana = None
Pozice = 5
Jazyk záznamu = GE

$ get_family_tree.py -id 15 Hledaná osoba:

ID = 15
Jméno = Maria - Marie
Pohlaví = F
Vyznání = catholic

Potomkovia: Marie (ID 15)
———————————————-
Prepojenie s osobou = Marie (ID 15) JE_MATKA Josef (ID 45)

ID = 45
Jméno = Josef - Josef
Pohlaví = M
Adresy = Adamsthal-Adamov 525
Nalezenec = 0

Rodiče: Marie (ID 15)

Prepojenie s osobou = Františka (ID 55) JE_MATKA Marie (ID 15)
ID = 55
Jméno = Franciska - Františka
Přímení = Šeda
Pohlaví = F

Prepojenie s osobou = Wenzel (ID 54) JE_OTEC Marie (ID 15)
ID = 54
Jméno = Wenzel
Přímení = Skuček
Pohlaví = M
Povolání = Zimmermans - Tesař
Adresy = in Chotzen-None