Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Tvorba rozsáhlých úložišť patentových dat

Plzeň 2022 Bc. Vojtěch Danišík

Místo této strany bude zadání práce.

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V diplomové práci jsou použity názvy programových produktů, firem apod., které mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

V Plzni dne 24. dubna 2022

Bc. Vojtěch Danišík

Poděkování

Děkuji panu Doc. Ing. Daliboru Fialovi, Ph.D. za ochotu při vedení diplomové práce a rady s jejím vypracováním.

Abstract

Creation of large-scale patent data repositories. The aim of the diploma thesis is to get acquainted with the available national sources of patent data and to create extensive local repositories of patent data enabling their effective searching and mining. The first part of the thesis thoroughly describes the types of patents, existing national sources and file formats in which patents are stored. Subsequently, the applicable technologies for searching and mining are described. The second part of the thesis is devoted to the selection of usable data and the implementation of selected technologies. Several queries and scenarios have been created to test efficient mining. The results of the testing are part of this work.

Abstrakt

Cílem diplomové práce je seznámit se s dostupnými národními zdroji dat o patentech a vytvořit rozsáhlá lokální úložiště patentových dat umožňující jejich efektivní prohledávání a vytěžování. První část práce důkladně popisuje typy patentů, existující národní zdroje a formáty souborů, ve kterých se patenty ukládají. Následně jsou popsány použitelné technologie pro prohledávání a vytěžování. Druhá část práce se věnuje výběru použitelných dat a implementaci vybraných technologií. Pro otestování efektivního vytěžování bylo vytvořeno několik query a scénářů. Výsledky testování jsou součástí této práce.

Obsah

1 Úvod

zmínit deployment - docker atp, aby se to mohlo narvat do teorie

2 Patent

todo

základní informace + uložiště dat + info o zdrojích (patentové úřady atp)

2.1 Patent vs Užitný vzor

2.2 Patent vs Průmyslový vzor

2.3 Patent vs Ochranná známka

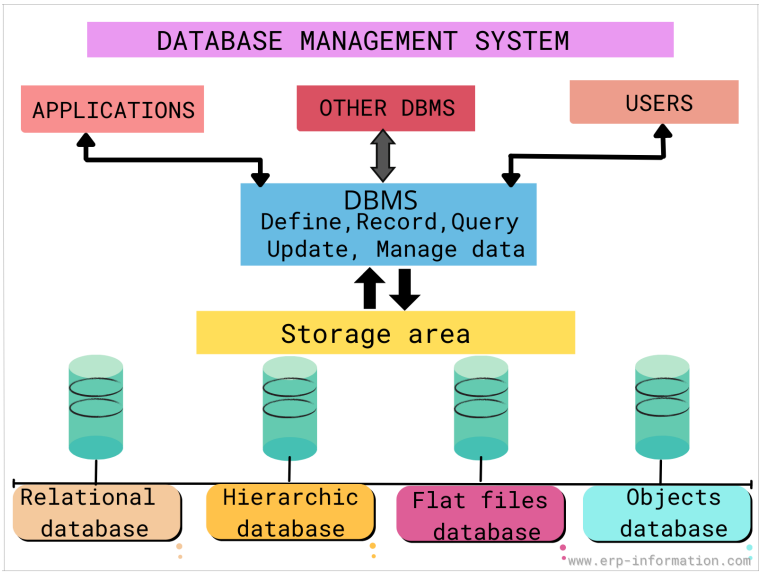
2.4 Patent vs Autorská práva

3 Databáze

Databáze je organizovaná kolekce strukturovaných informací nebo dat, která jsou typicky ukládána elektronicky v počítačovém systému. Data / informace lze nazvat jako fakta vztahující se k libovolnému uvažovanému objektu. Typický příklad objektu je člověk, jehož fakta jsou: jméno, věk, výška, váha a mnoho dalších [6].

3.1 Systém řízení báze dat

Pro správu dat v databázi a její řízení je potřeba komplexní software, který se nazývá **Systém Řízení Báze Dat** (SŘBD, anglicky DBMS). SŘBD slouží jako interface mezi samotnou databází a koncovým uživatelem (může být i program), umožňující jak vytěžování a aktualizaci dat, tak i možnosti nastavení záloh a jiných administrativních operací [1]. V dnešním světě existuje několik různých DBMS (například Relační DBMS, Objektově-orientované DBMS).



Obrázek 3.1: Systém řízení báze dat

3.2 Komponenty databáze

Všechny databáze sestávají z pěti základních komponent, nehledě na použitý typ databáze [5, 6]:

- **Hardware** - Fyzické stroje (počítače, servery, pevné disky, ...) na kterých běží databázový software.
- **Software** - Databázový software poskytuje uživateli / programu kontrolu nad databází. Zahrnuje to samotný databázový software, operační systém, software pro správu sdílení dat mezi uživately a programy pro přístup k datům v databázi.
- **Data** - Nezpracované a neorganizované fakty, které je potřeba zpracovat. Administrátor databáze organizuje tyto data a dává jim význam. Data se obecně skládají hlavně z faktů, observací, percepcí, čísel, znaků a mnoho dalších.
- **Jazyk** - Typický příklad použití jazyku je přístup k datům, přidávání nových dat, úpravu již existujících dat z databáze. Uživatel / program napíše specifické příkazy v jazyku pro přístup k datům (Database Access Language) a tyto příkazy následně pošle databázi ke zpracování. Více viz kapitola č. 3.3.
- **Procedury** - Procedura obsahuje předpřipravený seznam příkazů, které se následně vykonávají po zavolání dané procedury.

3.3 Jazyky

Databázové jazyky, jinak známé jako dotazovací jazyky, jsou klasifikací programovacích jazyků, které se používají k definování a přístupu k databázím. Pomocí těchto jazyků dokáže uživatel získávat nebo spravovat data v databázích. V dnešní době se jazyky (například SQL) mohou skládat ze čtyř podjazyků, kdy každý slouží k jinému účelu v rámci vykonávání příkazů [2, 7]:

- **Data definition language** (DDL) - DDL umí vytvářet jednotlivé komponenty databázového schématu (tabulky, soubory, indexy, ...), které tvoří strukturu reprezentující organizaci dat v databázi. Dostupné příkazy pro jazyk DDL:
  - **CREATE** - Vytvoření nového objektu (tabulka, index, ...).

  - **ALTER** - Změna struktury objektu.
  - **DROP** - Smazání objektu.
  - **RENAME** - Změna názvu objektu.
  - **TRUNCATE** - Smazání podobjektů v objektu (například záznamy v tabulce).

- **Data manipulation language** (DML) - DML slouží pro manipulaci s daty, které se nachází v již existující databázi. Dostupné příkazy pro jazyk DML:
  - **SELECT** - Získání záznamů (dat) z tabulky.
  - **INSERT** - Vložení nového záznamu (dat) do tabulky.
  - **UPDATE** - Úprava existujícího záznamu v tabulce.
  - **DELETE** - Smazání záznamu z tabulky.

- **Data control language** (DCL) - Pomocí DCL lze kontrolovat přístupy a práva k datům, které jsou uloženy v databázi. Uživateli lze nastavit práva k jednotlivým DML příkazům nad tabulkama / procedurama (například uživatel bude mít přístup pouze k příkazu SELECT nad tabulkou "TABULKA"). Dostupné příkazy pro jazyk DCL:
  - **GRANT** - Přidání práv uživateli nad danou tabulkou / procedurou.
  - **REVOKE** - Odebrání práv uživateli nad danou tabulkou / procedurou.

- **Transaction control language** (TCL) - TCL spravuje transakce v databázi. Transakce obsahuje jeden či více DML příkazů nad tabulkama, které se vykonávají po sobě. Všechny příkazy musí být úspěšně provedeny, aby bylo možné transakci označit za úspěšnou. Ukázka jedné transakce viz obrázek č. 3.2. Dostupné příkazy pro jazyk TCL:
  - **COMMIT** - Potvrzení transakce, změny provedené v transakci jsou permanentní a nejdou vzít zpět.
  - **ROLLBACK** - Vezme zpět veškerou práci v aktuální transakci. Lze se vrátit na začátek transakce nebo k SAVEPOINTu.
  - **SAVEPOINT** - Nastavení bodu v transakci, ke kterému se lze v budoucnu vrátit pomocí ROLLBACK.

```
SQL> SAVEPOINT SP1;
Savepoint created.
SQL> DELETE FROM CUSTOMERS WHERE ID=1;
1 row deleted.
SQL> SAVEPOINT SP2;
Savepoint created.
SQL> DELETE FROM CUSTOMERS WHERE ID=2;
1 row deleted.
SQL> SAVEPOINT SP3;
Savepoint created.
SQL> DELETE FROM CUSTOMERS WHERE ID=3;
1 row deleted.
SQL> ROLLBACK TO SP2;
Rollback complete.
```

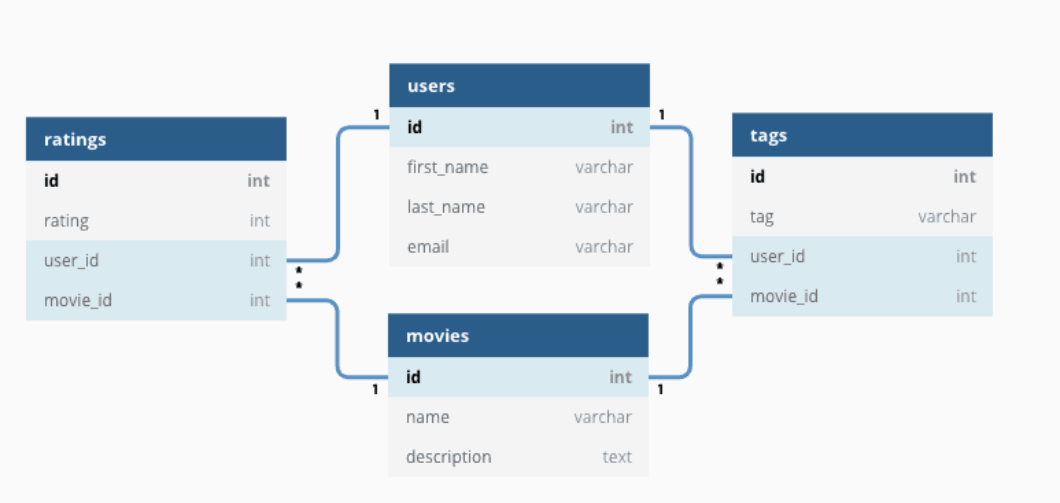
Obrázek 3.2: Ukázka jedné transakce (bez commitu)

3.4 Typy databází

V dnešním světě existuje mnoho různých typů databází. Výběr nejlepšího typu databáze pro konkrétní organizaci závisí na tom, jak organizace zamýšlí data používat. V této kapitole je vypsáno pouze pár typů, protože vznikají stále nové, méně známé typy databází, které jsou tvořeny pro specifické požadavky (například finanční, věděcké) [1, 3].

3.4.1 Relační databáze

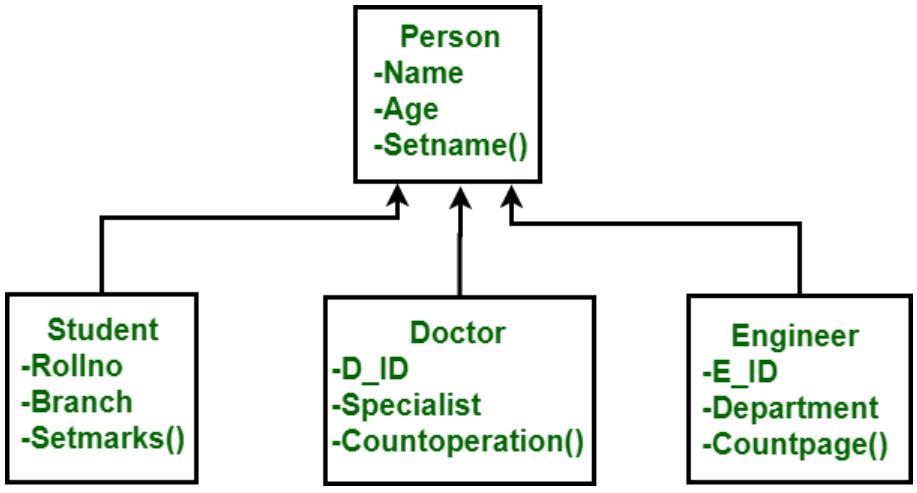
Název relační databáze pochází ze způsobu, jakým jsou data uložena, a to ve více souvisejících tabulkách. Data v tabulkách jsou uložena v řádcích a sloupcích. Relační databáze jsou velice spolehlivé a podporují všechny čtyři žádoucí vlastnosti databázových transakcí - ACID (Atomicita, Konzistence, Izolovanost, Trvalost). Pro co nejefektivěnjší využití tohoto typu databáze je potřeba ukládat pouze dobře strukturovaná data, pro částečně strukturovaná či nestrukturovaná data je vhodné použít například grafové nebo dokumentově založené databáze. Typické relační databáze jsou například: Microsoft SQL Server, Oracle Database, MySQL. Ukázku relační databáze lze vidět na obrázku č. 3.3.



Obrázek 3.3: Ukázka relační databáze

3.4.2 Objektově-orientovaná databáze

Objektově-orientovaná databáze je založena na objektově-orientovaném programování, kdy data a všechny jejich atributy a metody jsou svázány dohromady jako objekt. Stejně jako relační databáze, i objektově-orientované databáze odpovídají standardům ACID. Typické příklady jsou například: ObjectStore, Cache, ConceptBase. Ukázku objektově-orientované databáze lze vidět na obrázku č. 3.4.



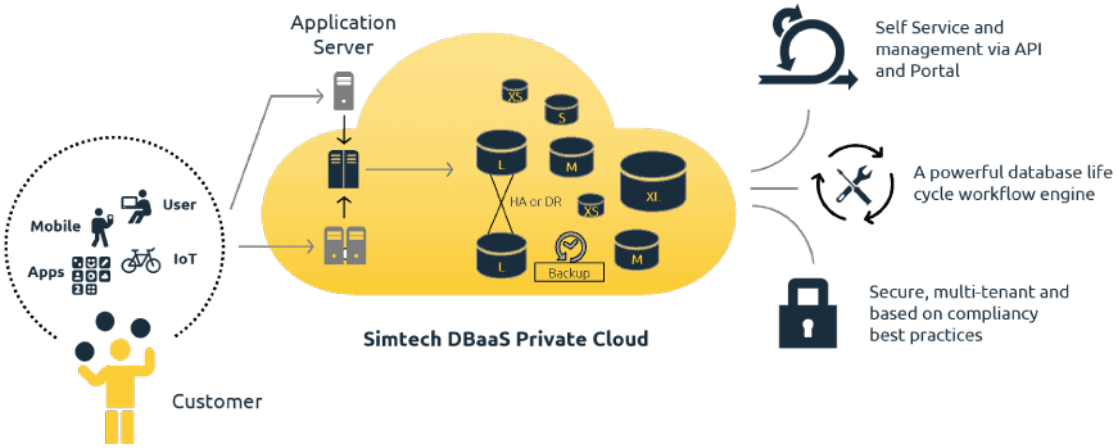
Obrázek 3.4: Ukázka objektově-orientované databáze

3.4.3 NoSQL databáze

todo

3.4.4 Cloud databáze

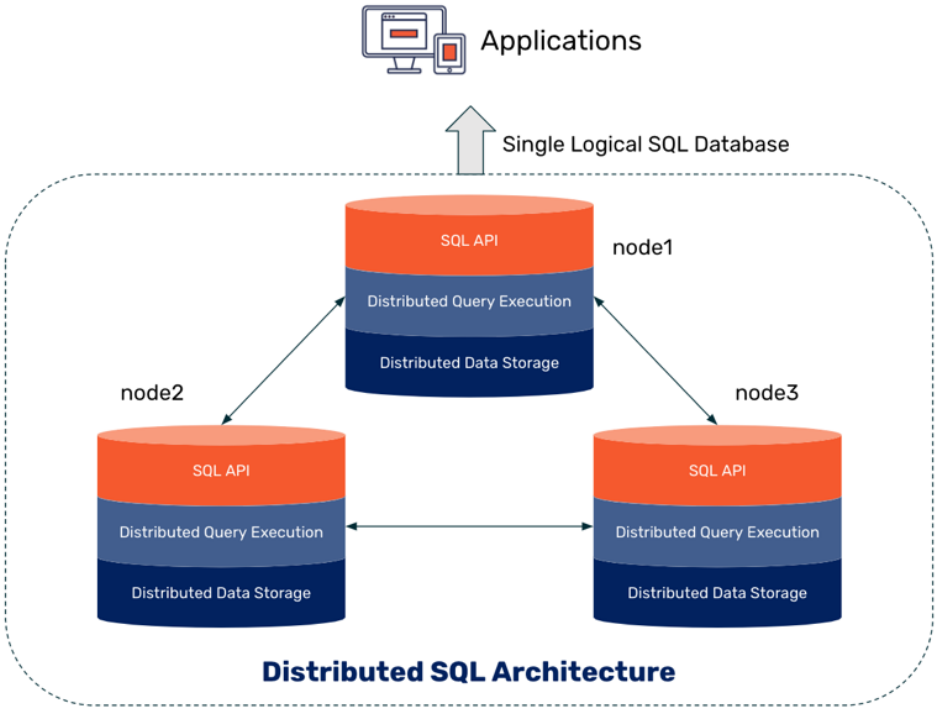
Cloudová databáze je kolekce dat, strukturovaných nebo nestrukturovaných, která se nachází na soukromé, veřejné nebo hybridní platformě cloud computingu. Stejně jako ostatní cloudové aplikace, cloudové databáze nabízejí flexibilitu a škálovatelnost spolu s vysokou dostupností. Existují dva typy modelů: tradiční a databázový jako služba (DBaaS). Typické příklady jsou: Amazon Web Service, Simtech, Google Cloud Platform. Ukázku cloud databáze lze vidět na obrázku č. 3.5.



Obrázek 3.5: Ukázka cloud databáze - databáze jako služba

3.4.5 Distribuovaná databáze

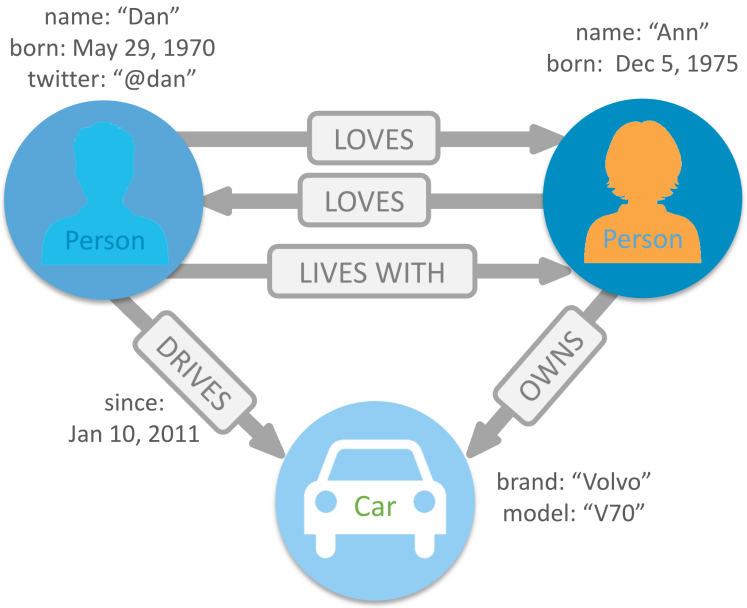
Distribuovaná databáze je skupina vzájemně propojených databází (dvě a více), které jsou fyzicky rozloženy na různých místech, která jsou spravována systémem správy distribuovaných databází (DBMS). V tomto typu systému nejsou data uložena na jednom místě. Při spuštění dotazu se pomocí kolektivní sady webů napříč různými datovými centry spolupracuje na zodpovězení otázky. Typické příklady jsou: Apache Ignite, AWS SimpleDB, FoundationDB. Ukázku distribuované databáze lze vidět na obrázku č. 3.6.



Obrázek 3.6: Ukázka distribuované databáze

3.4.6 Grafová databáze

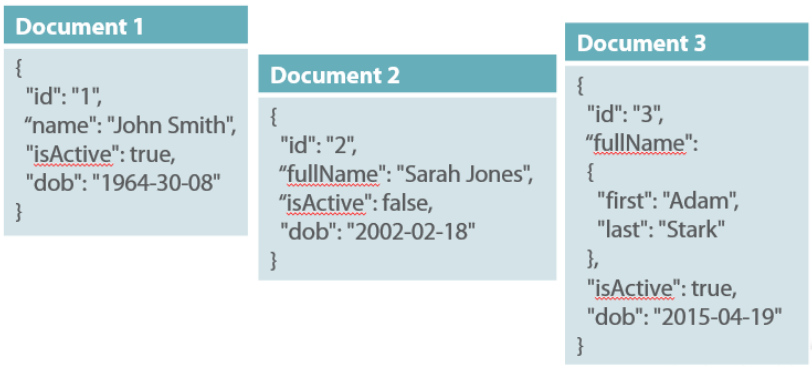
Grafová databáze je typem NoSQL databáze, která je založená na teorii grafů. Data jsou reprezentována jako uzly, hrany zase reprezentují vztahy mezi daty. Graf lze procházet podél určitých typů hran nebo přes celý graf. Procházení spojení nebo relací je velmi rychlé, protože vztahy mezi uzly se nepočítají v době dotazu, ale jsou v databázi trvalé. Typické příklady jsou: Neo4j, OrientDB, Microsoft Azure CosmosDB. Ukázku grafové databáze lze vidět na obrázku č. 3.7.



Obrázek 3.7: Ukázka grafové databáze

3.4.7 Dokumentová databáze

Databáze dokumentů jsou navržené pro ukládání, načítání a správu informací orientovaných na dokumenty. Dokumenty jsou obvykle uloženy ve formátu XML, JSON, BSON. Typické příklady jsou: MongoDB, Amazon DocumentDB, Apache CouchDB. Ukázku dokumentové databáze lze vidět na obrázku č. 3.8.

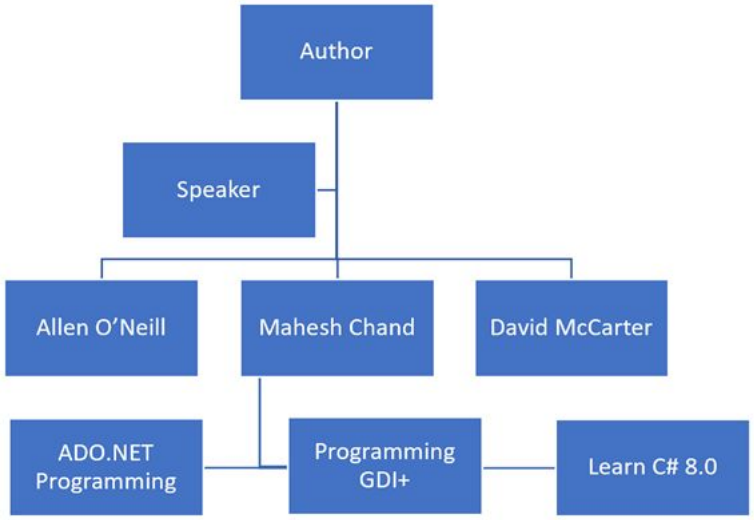


Obrázek 3.8: Ukázka dokumentově orientované databáze

3.4.8 Hierarchická databáze

Hierarchická databáze používají k ukládání dat model nadřazený-podřízený, který představuje data ve stromové podobě. Vztah mezi záznamy je jeden k

mnoha. To znamená, že jeden nadřazený uzel může mít mnoho podřízených, zatímco podřízený má vždy pouze jednoho nadřízeného.Typické příklady jsou: IBM Information Management System (IMS), RDM Mobile, Registry ve windows. Ukázku hierarchické databáze lze vidět na obrázku č. 3.9.



Obrázek 3.9: Ukázka hierarchické databáze

3.5 Dostupné databáze

3.5.1 MySQL

3.5.2 PostgreSQL

[4]Topten
[2]

4 Implementace

todo

4.1 Adresářová struktura

4.1.1 Patenty

4.1.2 Docker

4.2 Výběr dat

Zdroje (které ano, které ne + proč) + udělat stejnou tabulku jak v excelu + zmínit stránku ze který jsem čerpal informace (wipo.int) todo
Nezapomenout zmínit meze let, ve kterých se jednotlivé patenty z daných zemí nachází todo

4.2.1 Zdroje dat

Země	Patentový úřad	Zkratka
Anglie	Intellectual Property Office	IPO
Arménie	Intellectual Property Office	-
Austrálie	IP Australia	-
Bělorusko	National Center of Intellectual Property	NCIP
Bulharsko	Patent Office of Republic of Bulgaria	-
Česko	Industrial Property Office of the Czech Republic	-
Čína	China National Intellectual Property Administration	CNIPA
Dánsko	Danish Patent and Trademark Office	-
Egypt	Egyptian Patent Office	-
Estonsko	The Estonian Patent Office	-
Filipíny	Intellectual Property Office of the Philippines	IPOPHL
Finsko	Finnish Patent and Registration Office	PRH
Francie	National Institute of Industrial Property	INPI
Hong Kong	Intellectual Property Department	-
Chorvatsko	State Intellectual Property Office of the Republic of Croatia	SIPO
Indie	Office of the Controller General of Patents, Designs and Trade Marks	-
Indonésie	Directorate General of Intellectual Property	DGIP
Irsko	Intellectual Property Office of Ireland	IPOI
Island	Icelandic Intellectual Property Office	ISIPO
Israel	The Israel Patent Office	ILPO
Itálie	Directorate General for the Protection of Industrial Property	-
Japonsko	Japan Patent Office	JPO
Jižní Korea	Korean Intellectual Property Office	KIPO
Kanada	Canadian Intellectual Property Office	CIPO

Tabulka 4.1: Národní patentové úřady a jejich zkratky, část první

4.2.2 Atributy

Povinné atributy

Nepovinné atributy

4.2.3 Závěr výzkumu

Rovnou udělat kapitolu, ve který se aplikují všechny podmínky + se sepíše souhrn počtu patentů, z jakých zemí atp.

4.3 Implementace databáze

4.3.1 MySQL

4.3.2 MongoDB

4.3.3 ElasticSearch

4.4 Výsledný modul

4.4.1 Technologické požadavky

4.4.2 Docker

Images, scripty
Import dat (skripty, data soubory, ...)

4.4.3 Inicializace MySQL

4.4.4 Inicializace MongoDB

Země	Patentový úřad	Zkratka
Kuba	Cuban Industrial Property Office	OCPI
Litva	State Patent Bureau of the Republic of Lithuania	-
Lotyšsko	Patent Office of the Republic of Latvia	-
Maďarsko	Hungarian Intellectual Property Office	HIPO
Malajsie	Intellectual Property Corporation of Malaysia	MyIPO
Mexiko	Instituto Mexicano De La Propiedad Industrial	IMPI
Moldova	State Agency on Intellectual Property	AGEPI
Německo	German Patent and Trade Mark Office	DPMA
Nizozemsko	Netherlands Patent Office	-
Norsko	Norwegian Industrial Property Office	NIPO
Nový Zéland	Intellectual Property Office of New Zealand	IPONZ
Peru	National Institute for the Defense of Competition and Protection of Intellectual Property	INDECOPI
Polsko	Urząd Patentowy Rzeczypospolitej Polskiej	UPRP
Portugalsko	Portuguese Institute of Industrial Property	-
Rakousko	Austrian Patent Office	-
Rumunsko	State Office for Inventions and Trademarks	OSIM
Rusko	Federal Service for Intellectual Property	Rospatent
Řecko	Hellenic Industrial Property Organization	HIPO
Singapur	Intellectual Property Office of Singapore	IPOS
Slovensko	Industrial Property Office of the Slovak Republic	-
Slovinsko	Slovenian Intellectual Property Office	SIPO
Srbsko	Intellectual Property Office of the Republic of Serbia	-
Španělsko	Spanish Patent and Trademark Office	OEPM
Švédsko	Swedish Intellectual Property Office	PRV
Švýcarsko	Swiss Federal Institute of Intellectual Property	-
Turecko	Turkish Patent and Trademark Office	Turkpatent
Ukrajina	Ukrainian Intellectual Property Institute	Ukrpatent

Tabulka 4.2: Národní patentové úřady a jejich zkratky, část druhá

Země	Název patentu	Rok přihlášky / patentu	Autor	ID patentu
Kanada	x	x	x	x
Česko	x	x	-	x
Litva	x	x	x	x
Portugalsko	x	x	x	x
Španělsko	x	x	x	x
Švédsko	-	x	-	x
Izrael	x	x	x	x
Itálie	x	x	x	x
Mexiko	x	x	x	x
Polsko	x	x	-	-
Anglie	x	x	x	x
Rusko	x	x	x	x
Peru	x	x	x	x
Francie	x	x	x	x

Tabulka 4.3: Povinné atributy nacházející se v dostupných patentech

Země	Abstrakt	Slovník	Reference	Žadatel
Kanada	-	-	-	x
Česko	x	-	x	-
Litva	x	-	-	x
Portugalsko	x	-	-	x
Španělsko	x	-	-	x
Švédsko	x	-	-	-
Izrael	-	-	-	x
Itálie	-	-	-	x
Mexiko	x	-	-	x
Polsko	x	x	-	-
Anglie	-	-	-	-
Rusko	-	-	-	-
Peru	-	-	-	-
Francie	x	-	x	x

Tabulka 4.4: Nepovinné atributy nacházející se v dostupných patentech, část první

Země	**Adresa**	**Rodina patentů**	**Obor**	**Fulltext**
Kanada	x	-	x	-
Česko	-	-	x	-
Litva	-	-	x	-
Portugalsko	-	-	x	-
Španělsko	x	-	x	x
Švédsko	-	-	x	x
Izrael	x	-	-	-
Itálie	-	-	-	-
Mexiko	-	-	x	-
Polsko	-	-	-	-
Anglie	-	-	x	-
Rusko	-	-	-	-
Peru	-	-	x	-
Francie	-	-	x	x

Tabulka 4.5: Nepovinné atributy nacházející se v dostupných patentech, část druhá

5 Rozšiřitelnost modulu

Zadání diplomové práce sice splněno bylo, ale v blízké budoucnosti mohou být požadavky na modul změněny. Jako příklad lze uvést podporu přidávání nových patentů do databází, zjištění autorů pro české patenty, automatické stahování dat z již ověřených patentových zdrojů. V této kapitole jsou popsány 3 možné návrhy na rozšíření modulu ohledně importu dat do již existujících databází.

5.1 Přidávání nových patentů

Cílem tohoto rožšíření by bylo automatické přidávání patentů z datových souborů jak do MySQL databáze, tak i do Mongo.

Rozšíření by se dalo realizovat jako aplikace ve vyšším programovacím jazyku (např. Java, C), kdy vstupem do aplikace by byl soubor v datovém formátu JSON/XML/CSV a jiné. Vstupní soubor by se následně:

- převedl na JSON řetězec (v případě že soubor není ve formátu JSON) a vložil do Mongo databáze
- rozparsoval a extrahovali by se všechny atributy, které se ukládají v MySQL databázi (viz mysql kapitola) TODO

Jelikož je dost časté, že každý národní zdroj dat používá odlišnou strukturu patentu, tak bude potřeba aplikaci neustále upravovat (ať už v rámci přidávání nových zdrojů, nebo v případě změny struktury patentu u již podporovaných zdrojů).

Jako další velký problém lze zmínit extrakci atributů patentu ze souborů. Tím, že různé patentové soubory mají odlišnou strukturu, to znamená hloubku zanoření specifických elementů, jiné názvy elementů, tak bude obtížné naimplementovat řešení extrakce pro všechny soubory. Tento problém by se dal řešit tak, že se vytvoří soubory se slovníkama, které by obsahovali názvy elementů pro daný atribut. Slovníky by se následně použily při extrakci.

5.2 Zjišťování autorů pro české patenty

Český národní patentový úřad poskytuje data o českých patentech, které ale neobsahují autora ani instituci. Pro zjištění autora nebo instituce, která

patent registrovala, je nutné použít oficiální vyhledávač. Cílem tohoto rozšíření by bylo vytvořit aplikaci ve vyšším programovacím jazyku, která se pro všechny české patenty bude snažit najít jejich autory za pomoci využití prohledávačů webů (web crawler). Postupů řešení může být mnoho:

- Zjišťování autorů by se provedlo pro všechny existující české patenty v databázi. Z MySQL databáze se zjistí všechny ID patentů pro české patenty, které se následně použijí jako vstup pro web crawler.
- Zjišťování autorů by se provedlo pro patent/y uložené v souboru, kdy aplikace by pro všechny patenty v souboru zjistila autory a následně je dopsala do příslušnýho elementu patentu v daném souboru.
- Stejný postup jako předchozí s tím rozdílem, že po zjištění autora se patent rovnou přidá do MySQL i Mongo databáze.

5.3 Automatické stahování dat z ověřených zdrojů

Cílem tohoto rozšíření by bylo automatické stahování dat (případně i jejich parsování) z ověřených zdrojů. Ověřené zdroje by byly uloženy například v XML souboru, kdy každý zdroj by měl tyto položky:

- **Název země**
- **URL** - URL zdroje dat, na které lze stáhnout data.
- **XPath** - XPath výraz, pomocí kterého lze ze stránky vyfiltrovat a získat odkazy ke stažení dat
  (např. */html/body//a[contains(@href,'example')]/@href*)
- **Poslední verze** - Název / číslo poslední stažené verze.

XML soubor by byl následně zpracován pomocí aplikace (např. Java, C#), která by následně pro každý zdroj dat provedla následující kroky:

1. Získání seznamu odkazů na zdroje dat.
2. Stažení všech zdrojů dat, jejichž verze je větší než aktuálně uložená verze v XML.
3. V tomto bodě se dá naimplementovat cokoliv - např. lze uložená data extrahovat ze ZIP souborů, importovat patenty do databází (viz kapitola č. 5.1), pouze notifikace o stažení několika nových souborů z daty a mnoho dalšího.

4. Aktualizace verze v XML souboru.

Automatizace stahování dat by spočívala ve spouštění aplikace pro stahování dat v pravidelných intervalech (např. každé druhé úterý v 17:00). Jako příklad lze uvést použití pipeline na Jenkins serveru, který bude spouštět z lokálního uložiště spustitelnou aplikaci v daný čas (pomocí CRON). Po vykonání celého procesu může Jenkins poslat email o stavu posledního spuštění (zda se spuštění povedlo, kolik souborů byl schopen stáhnout pro jaké země, ...). Samozřejmě bohatě postačí i použití plánovače v OS.

6 Ověření efektivního vytěžování

K ověření efektivního vytěžování bylo připraveno několik scénářů jak pro SQL, tak i pro Mongo + ElasticSearch.
Napsat referenční stroj na kterém se testovalo - CPU, RAM, ... todo

6.1 Mongo + ElasticSearch

6.2 MySQL

Pro MySQL bylo připraveno 10 scénářů, které testují všechny vytvořené tabulky v databázi. Každý scénář obsahuje textový popis, SQL příkaz, rychlost vykonání příkazu a ukázku výsledků.

6.2.1 Scénář č.1

Textový popis: Pět nejčastěji patentujících institucí v Izraeli v roce 2015
SQL:

```
select count(*), inventors.inventor from inventors
   left outer join patents on inventors.id_patent =
   patents.id where YEAR(patents.patent_date) = 2015
   and patents.patent_id like '%IL%' group by inventors
   .inventor order by count(*) desc LIMIT 5;
```

Rychlost vykonání dotazu: TODO
Výsledek dotazu:

count(*)	inventor
39	DOW AGROSCIENCES LLC
29	F. HOFFMANN-LA ROCHE AG
29	GENENTECH, INC.
29	NOVARTIS AG
24	RAYTHEON COMPANY

Obrázek 6.1: Ukázka výsledku dotazu pro scénář č.1

6.2.2 Scénář č.2

Textový popis: Tři nejméně patentované obory v Kanadě od roku 2010
SQL:

```
select count(*), classification.section from
    classification left outer join patents on patents.id
     = classification.id_patent where YEAR(patents.
    patent_date) >= 2010 and patents.patent_id like '%CA
    %' group by classification.section order by count(*)
     asc LIMIT 3;
```

Rychlost vykonání dotazu: TODO
Výsledek dotazu: TODO

count(*)	inventor
8	Gould Nigel
4	Philips Andrew
4	Lee Sangik

Obrázek 6.2: Ukázka výsledku dotazu pro scénář č.2

6.2.3 Scénář č.3

Textový popis: Nejčastější klasifikace patentu za rok 2008 ve Španělsku
SQL:

```
select count(*), classification.section,
    classification.class, classification.subclass from
    classification left outer join patents on patents.id
     = classification.id_patent where YEAR(patents.
    patent_date) = 2008 and patents.patent_id LIKE '%ES%
    ' group by classification.section, classification.
    class, classification.subclass order by count(*)
    desc LIMIT 1;
```

Rychlost vykonání dotazu: TODO
Výsledek dotazu:

count(*)	section	class	subclass
73	B	65	D

Obrázek 6.3: Ukázka výsledku dotazu pro scénář č.3

6.2.4 Scénář č.4

Textový popis: Autor s největším počtem patentů ze všech zemí
SQL:

```
select count(*), inventors.inventor from inventors
    left outer join patents on patents.id = inventors.
    id_patent group by inventors.inventor order by count
    (*) desc LIMIT 1;
```

Rychlost vykonání dotazu: TODO
Výsledek dotazu: TODO

count(*) 1	inventor
8	Gould Nigel
4	Philips Andrew
4	Lee Sangik

Obrázek 6.4: Ukázka výsledku dotazu pro scénář č.4

6.2.5 Scénář č.5

Textový popis: Nejméně používaný jazyk pro patenty za rok 2003
SQL:

```
select count(*), patents.language from patents where
    patents.language not like '%-%' group by patents.
    language order by count(*) asc LIMIT 1;
```

Rychlost vykonání dotazu: TODO
Výsledek dotazu: TODO

6.2.6 Scénář č.6

Textový popis: Deset Institucí / autorů s patenty pokrývající největší množství oborů
SQL:

count(*)	inventor
8	Gould Nigel
4	Philips Andrew
4	Lee Sangik

Obrázek 6.5: Ukázka výsledku dotazu pro scénář č.5

```
select count(distinct classification.section),
   inventors.inventor from inventors left outer join
   classification on classification.id_patent =
   inventors.id_patent where section is not null group
   by inventors.inventor order by count(distinct
   classification.section) desc LIMIT 10;
```

Rychlost vykonání dotazu: TODO
Výsledek dotazu: TODO

count(*)	inventor
8	Gould Nigel
4	Philips Andrew
4	Lee Sangik

Obrázek 6.6: Ukázka výsledku dotazu pro scénář č.6

6.2.7 Scénář č.7

Textový popis: Země s nejvíce patenty od roku 2018
SQL:

```
select count(*), patents.country from patents where
   YEAR(patents.patent_date) >= 2018 group by patents.
   country order by count(*) desc;
```

Rychlost vykonání dotazu: TODO
Výsledek dotazu: TODO

count(*) ▼ 1	inventor
8	Gould Nigel
4	Philips Andrew
4	Lee Sangik

Obrázek 6.7: Ukázka výsledku dotazu pro scénář č.7

6.2.8 Scénář č.8

Textový popis: Nejvíce používaný jazyk pro patenty ve Francii
SQL:

```
select count(*), patents.language from patents where
    patents.patent_id like '%FR%' group by patents.
    language order by count(*) desc;
```

Rychlost vykonání dotazu: TODO
Výsledek dotazu: TODO

count(*) ▼ 1	inventor
8	Gould Nigel
4	Philips Andrew
4	Lee Sangik

Obrázek 6.8: Ukázka výsledku dotazu pro scénář č.8

6.2.9 Scénář č.9

Textový popis: Tři nejčastěji patentující instituce / autoři v Anglii v textilním oboru za rok 2013
SQL:

```
select count(*), inventors.inventor from inventors
    left outer join patents on patents.id = inventors.
    id_patent left outer join classification on
    classification.id_patent = patents.id where
    classification.section like '%D%' and patents.
    patent_id like '%GB%' and YEAR(patents.patent_date)
    = 2013 group by inventors.inventor order by count(*)
     desc LIMIT 3
```

Rychlost vykonání dotazu: TODO
Výsledek dotazu:

count(*) ▼ 1	inventor
8	Gould Nigel
4	Philips Andrew
4	Lee Sangik

Obrázek 6.9: Ukázka výsledku dotazu pro scénář č.9

7 Závěr

Literatura

[1] *What Is a Database?* [online]. Oracle. [cit. 21.04.2022]. Dostupné z: https://www.oracle.com/database/what-is-database/.

[2] *Types of Database Languages and Their Uses (Plus Examples)* [online]. indeed, 2021. [cit. 21.04.2022]. Dostupné z: https://www.indeed.com/career-advice/career-development/database-languages.

[3] *The Types of Databases (with Examples)* [online]. Matillion, 2018. [cit. 22.04.2022]. Dostupné z: https://www.matillion.com/resources/blog/the-types-of-databases-with-examples.

[4] Kamaruzzaman, M. *Top 10 Databases to Use in 2021* [online]. Towards Data Science, 2021. [cit. 21.04.2022]. Dostupné z: https://towardsdatascience.com/top-10-databases-to-use-in-2021-d7e6a85402ba.

[5] Lutkevich, B. *database (DB)* [online]. Tech Target, 2021. [cit. 21.04.2022]. Dostupné z: https://www.techtarget.com/searchdatamanagement/definition/database.

[6] Peterson, R. *What is a Database? Definition, Meaning, Types with Example* [online]. Guru99, 2022. [cit. 21.04.2022]. Dostupné z: https://www.guru99.com/introduction-to-database-sql.html.

[7] Singh, C. *DBMS languages* [online]. BeginnersBook, 2015. [cit. 21.04.2022]. Dostupné z: https://beginnersbook.com/2015/04/dbms-languages/.

A Uživatelská dokumentace

B Vzhled modulů