

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačních technologií**



**Bakalářská práce**

**Registr platební morálky**

**Tomáš Rosa**

© 2021 ČZU v Praze

# ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Tomáš Rosa

Ekonomika a management  
Provoz a ekonomika

Název práce

Registr platební morálky

Název anglicky

Payment behavior register

---

### Cíle práce

Bakalářská práce je tematicky zaměřena na problematiku registru platební morálky. Hlavním cílem práce je analýza technologií a možností realizace registru platební morálky s realizací prototypu. Dílčím cíli práce jsou

- vypracování přehledu již existujících registrů
- vypracování přehledu technologií pro realizaci registrů.

### Metodika

Metodika řešení problematiky bakalářské práce je založena na studiu a analýze odborných informačních zdrojů. Vlastní práce spočívá v analýze technologií a řešení registrů s realizací prototypu. Na základě syntézy teoretických poznatků a výsledků praktické části budou formulovány závěry bakalářské práce.

**Doporučený rozsah práce**

60 stran textu.

**Klíčová slova**

registr, big data, databáze, spotřebitel, informace, platby, platební morálka

---

**Doporučené zdroje informací**

HOLUBOVÁ, Irena, Jiří KOSEK, Karel MINAŘÍK a David NOVÁK. Big Data a NoSQL databáze. Praha: Grada, 2015. Profesionál. ISBN 978-80-247-5466-6.

MEJSTRÍK, Michal, Magda PEČENÁ a Petr TEPLÝ. Bankovníctví v teorii a praxi: Banking in theory and practice. Praha: Karolinum, 2014. ISBN 978-80-246-2870-7.

ONDŘEJ, Jan. Spotřebitelské smlouvy a ochrana spotřebitele: ekonomické, právní a sociální aspekty. Praha: C.H. Beck, c2013. Beckova edice právní instituty. ISBN 978-80-7400-446-9.

Sbírka zákonů Česká republika. Břeclav: Moraviapress. ISSN 1211-1244.

TYRYCHTR, J. Provozní a analytické databáze – Teoretické základy. Praha : ČSVIZ, 2015. ISBN 978-80-87968-02-4.

---

**Předběžný termín obhajoby**

2020/21 LS – PEF

**Vedoucí práce**

Ing. Pavel Šimek, Ph.D.

**Garantující pracoviště**

Katedra informačních technologií

---

Elektronicky schváleno dne 29. 7. 2020

Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

---

Elektronicky schváleno dne 19. 10. 2020

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 09. 03. 2021

### **Čestné prohlášení**

Prohlašuji, že svou bakalářskou práci "Registr platební morálky" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 15.03.2021

---

### **Poděkování**

Rád bych touto cestou poděkoval panu inženýrovi Pavlu Šimkovi, za možnost vykonat práci pod jeho vedením.

# Registr platební morálky

## Abstrakt

Práce pojednává o problematice finančních registrů na území České republiky. V první části se zabývá již existujícími registry a právními subjekty, které jsou s nimi spjaty. Na ni navazuje rozbor právní normy, která může být, pro vytvoření registru užita. V další části jsou popsány existující technologie spolu s vybranými zástupci těchto technologií. Z popsaných technologií je posléze, na základě jejich parametrů, vybrána odpovídající sada řešení. Ta je využita při tvorbě databázového prototypu, který je rovněž součástí bakalářské práce. Dále jsou popsány funkce prototypu a postup, podle kterého je možné jej spustit. V poslední části se práce zabývá polemikou dalších možných řešení a budoucím vývojem.

**Klíčová slova:** registr, big data, databáze, spotřebitel, informace, platby, platební morálka, SQL, úvěry

# Payment behavior register

## **Abstract**

This thesis speaks about the problematic of financial registers in the Czech Republic. In the first part, existing registers and companies connected with those are being discussed. Next, legal norms which can be applied for such a register are analysed. In the following part, existing technologies including their representatives are described. The right solution is being chosen from these technologies according to their specifications. This solution is used then for the realization of a database prototype which is also part of the thesis. Following, functions of the prototype and the process of its setting is being described. In the last part, the thesis argues about other possible solutions and about further development.

**Keywords:** register, big data, database, consumer, information, payments, payment behavior, SQL, credits

# Obsah

<b>1 Úvod.....</b>	<b>11</b>
<b>2 Cíl práce a metodika .....</b>	<b>12</b>
2.1 Cíl práce .....	12
2.2 Metodika .....	12
<b>3 Teoretická východiska .....</b>	<b>13</b>
3.1 Komerční subjekty spjaté s registry .....	13
3.1.1 SOLUS, zájmové sdružení právnických osob .....	13
3.1.2 CRIF - Czech Credit Bureau, a.s. ....	14
3.2 Úvěrové Registry .....	14
3.2.1 Centrální Registr Úvěrů (CRÚ) .....	14
3.2.2 Registr Fyzických Osob (Registr FO) .....	15
3.2.3 Registr podnikatelů a právnických osob (Registr IČ).....	16
3.2.4 Pozitivní registr .....	17
3.2.5 Bankovní Registr Klientských Informací (BRKI) .....	18
3.2.6 Nebankovní Registr Klientských Informací (NRKI).....	19
3.2.7 Registr Platebních Informací (REPI).....	21
3.3 Právní rámec a zpracování osobních údajů .....	22
3.3.1 Zákon o ochraně spotřebitele .....	22
3.3.2 Zákon o bankách .....	24
3.4 Databáze .....	25
3.4.1 Historie.....	25
3.5 Databázové modely .....	26
3.5.1 Hierarchické databáze .....	26
3.5.2 Síťové databáze .....	27
3.5.3 Relační databáze .....	27
3.5.4 Objektové databáze .....	28
3.6 Databázové systémy .....	28
3.6.1 PostgreSQL .....	29
3.6.2 MySQL .....	30
3.6.3 Oracle Database .....	30
3.6.4 Microsoft SQL Server .....	31
3.6.5 MongoDB .....	32
3.7 Databázové Frameworky .....	33
3.7.1 Spring framework .....	33
3.7.2 Micronaut.....	34
3.7.3 Hibernate.....	34



3.8	Metoda TOPSIS .....	35
<b>4</b>	<b>Vlastní práce .....</b>	<b>36</b>
4.1	Požadavky na prototyp .....	36
4.2	Databázové Frameworky .....	37
4.2.1	Spring .....	37
4.2.2	Micronaut .....	39
4.2.3	Hibernate .....	40
4.3	Databázové systémy .....	40
4.3.1	MySQL .....	42
4.3.2	Oracle Database .....	43
4.3.3	Microsoft SQL Server .....	44
4.3.4	PostgreSQL .....	44
4.3.5	MongoDB .....	45
4.4	Vícekritériální analýza variant .....	46
<b>5</b>	<b>Výsledky a diskuse .....</b>	<b>49</b>
5.1	Diskuse .....	50
<b>6</b>	<b>Závěr .....</b>	<b>51</b>
<b>7</b>	<b>Seznam použitých zdrojů .....</b>	<b>52</b>

## Seznam tabulek

Tabulka 1	Kriteriální matice frameworků .....	46
Tabulka 2	Normalizovaná matice frameworků .....	47
Tabulka 3	Kriteriální matice databází .....	47
Tabulka 4	Normalizovaná matice databází .....	48
Tabulka 5	Výpočet pořadí frameworků .....	49
Tabulka 6	Výpočet pořadí databází .....	49

## Seznam použitých zkratk

- ACID Atomic, Consistent, Isolated, Durable
- API Application Programming Interface
- BRKI Bankovní registr klientských informací
- CERN Evropská organizace pro jaderný výzkum
- CRUD Create, Read, Update, Delete
- CRÚ Centrální registr úvěrů
- ČNB Český národní banka

- ETL Extract, Transform, Load
- EU Evropská Unie
- FO Fyzická osoba
- FOP Fyzická osoba podnikatel
- GIS Geografický informační systém
- IBM International Business Machines Corporation
- IČO Identifikační číslo osoby
- IMS IBM Information management System
- JPA Java Persistence API
- JSON JavaScript Object Notation
- JVM Java Virtual Machine
- JWT JSON Web Token
- NoSQL Non Structured Query Language
- NRKI Nebankovní registr clientských informací
- ORM Objektově relační mapování
- PO Právnícká osoba
- RAM Random Access Memory
- REPI Registr platebních informací
- REST Representational State Transfer
- SEQUEL Structured English Query Language
- SOLUS Sdružení na Ochranu Leasingu a Úvěrů Spotřebitelům
- SQL Structured Query Language
- USA Spojené státy Americké

# 1 Úvod

Situace na přelomu let 2019 a 2020 napovídala o zpomalení ekonomiky, které bylo ještě umocněno rozšířením onemocnění COVID-19, respektive vyhlášením opatření, kvůli kterým došlo k uzavření obchodů.

Na možnosti zpomalení upozorňoval, na počátku roku 2020, ve svém článku, v periodiku Bankovníctví, výkonný ředitel CRIF – Czech Credit Bureau, a.s. Petr Kučera.

Dle (Kučera, 2020, s.34) to vše (zpomalení ekonomiky, růst objemu poskytnutých úvěrů a pokles hodnoty takzvaného ohroženého dluhu) může přinést vyšší delikvenci při splácení úvěrů a současně vést některé spotřebitele k větší ochotě pouštět se do sporů ohledně neplatnosti úvěrové smlouvy v případech, kdy byly úvěry sjednané bez dostatečného prověření úvěruschopnosti klienta.

To se stalo důvodem pro zvýšení zájmu o registry a databáze, které disponují informacemi o úvěrových historiích klientů uživatelů registru.

V současné době jsou provozovateli obsahově největších úvěrových registrů na území České republiky Česká národní banka, zájmové sdružení právnických osob SOLUS, CBCB – Czech Banking Credit Bureau, a. s., CNCB – Czech Non-Banking Credit Bureau, z.s.p.o. a CRIF - Czech Credit Bureau, a.s.

Důležitým aspektem spjatým s registry a databázemi jsou technologie, které jsou užity pro jejich tvorbu a provoz. Technologie jsou vybírány na základě předem určených parametrů, které vyplývají z požadavků na samotnou databázi.

Jedním z nejdůležitějších faktorů je účel. Zadavatel by měl mít alespoň základní představu o zaměření databáze a o skupině uživatelů, pro kterou je určena.

V současnosti lze databáze vybudovávat od základů, což v mnoha případech není z finančního hlediska doporučováno nebo využívat částí již připravených technologií, jako příklad lze uvést společnost Oracle poskytující databázové systémy.

V neposlední řadě v případě sdílení informací, musí zadavatel, případně provozovatel, dbát udržení bezpečnosti a na dodržování platných zákonů. Zákony a nařízení, jako například zákon o ochraně spotřebitele nebo obecné nařízení o ochraně osobních údajů (GDPR), vymezují kdo, jakým způsobem a s jakými daty může nakládat.

Registry, a to hlavně finanční, které vytváří modely s popisem schopnosti a ochoty spotřebitele splnit jeho smluvní závazky, musí umět rychle zareagovat na změny, aby nedošlo k poškození spotřebitele. (Sbírka zákonů České republiky)

## **2 Cíl práce a metodika**

### **2.1 Cíl práce**

Bakalářská práce je tematicky zaměřena na problematiku registru platební morálky. Hlavním cílem práce je analýza technologií a možností realizace registru platební morálky s realizací prototypu. Dílčími cíli práce jsou

- vypracování přehledu již existujících registrů
- vypracování přehledu technologií pro realizaci registrů.

### **2.2 Metodika**

Metodika řešené problematiky bakalářské práce je založena na studiu a analýze odborných informačních zdrojů. Vlastní práce spočívá v analýze technologií a řešení registrů s realizací prototypu. Na základě syntézy teoretických poznatků a výsledků praktické části budou formulovány závěry bakalářské práce. Pro analýzu technologií je užito metody vícekriteriální analýzy. Metoda TOPSIS, jedna z forem vícekriteriální analýzy, funguje na principu minimalizace vzdálenosti od ideální varianty. Výpočty pro určení adekvátního řešení byly realizovány za pomoci programu MS Excel.

### **3 Teoretická východiska**

I přes finanční zaměření bakalářské práce je třeba zmínit, že v České republice existuje nepřehledné množství registrů a evidencí. Mezi takové patří například registr vozidel, registr smluv nebo centrální evidence exekucí, které jsou provozovány státem nebo státními institucemi a, jak již názvy napovídají, jsou užívány v různých odvětvích.

Úlohou úvěrových registrů je zabránit předlužování klientů a zlepšit informovanost finančních institucí o úvěrové historii jejich stávajících nebo potenciálních klientů. Tedy úvěrové registry by měly být užívány jako jeden z pomocných rozhodovacích nástrojů, které mají finanční instituce k dispozici.

V České republice, na rozdíl od jiných zemí světa, nejsou informace o platební morálce a schopnostech fyzických osob (FO) shromažďovány na státní úrovni. Příklady států, které data o fyzických osobách shromažďují jsou Francie, Maďarsko nebo Rusko.

Stát, respektive Česká národní banka (ČNB) provozuje Centrální Registr Úvěrů (CRÚ), který je ovšem zaměřen na fyzické osoby podnikatele (FOP) a právnické osoby (PO). (Centrální registr úvěrů, 2021)

To umožnilo komerčním subjektům, mezi které patří hlavně Sdružení na Ochranu Leasingu a Úvěrů Spotřebitelům (SOLUS) a CRIF - Czech Credit Bureau, aby odpověděly na tuto poptávku, ve formě dat o fyzických osobách (FO), svými produkty.

#### **3.1 Komerční subjekty spjaté s registry**

##### **3.1.1 SOLUS, zájmové sdružení právnických osob**

SOLUS neboli Sdružení na Ochranu Leasingu a Úvěrů Spotřebitelům vzniklo v červnu roku 1999 jako zájmové sdružení právnických osob, které mezi sebou byly ochotné sdílet data o svých klientech. (SOLUS - Zájmové sdružení právnických osob, 2021)

Postupem času se ke sdružení připojily subjekty z odvětví bankovníctví, telekomunikací, energetiky a pojišťoven. V současnosti je do sdružení zapojeno přes padesát subjektů napříč výše zmíněnými odvětvími.

Sdružení spravuje celkem šest registrů, z nichž jsou pouze tři úvěrové. Jedná se o Registr Fyzických Osob, Registr podnikatelů a právnických osob, Pozitivní registr, Registr třetích stran, a dva pomocné registry, Registr neplatných dokladů a Insolvenční registr.

### **3.1.2 CRIF - Czech Credit Bureau, a.s.**

Společnost CRIF – Czech Credit Bureau vznikla v roce 2005, kdy původně italská společnost CRIF odkoupila 100% akcií firmy CCB – Czech Credit Bureau. Společnost poskytuje technickou podporu tří registrům zaměřeným právě na platební informace subjektů, kterými jsou klienti uživatelů registrů. (CRIF - Czech Credit Bureau, 2021)

## **3.2 Úvěrové Registry**

V současné době můžeme hovořit o sedmi úvěrových registrech, které jsou na území České republiky provozovány. Jedná se o Centrální Registr Úvěrů, Registr Fyzických Osob, Registr podnikatelů a právnických osob, Pozitivní registr, Bankovní Registr Klientských Informací, Nebankovní Registr Klientských Informací a Registr Platebních Informací.

Informace v registrech se rozdělují na pozitivní a negativní. Zjednodušeně lze říci, že v případě negativních informací hovoříme o závazcích (platbách, kontraktech a smlouvách), které nejsou řádně hrazeny nebo jsou v určitém stupni splatnosti případně po splatnosti. Jako pozitivní jsou pak brána data, která hovoří o správné platební morálce subjektu.

Takovéto informace lze získat z různých druhů reportů z daných registrů, které principiálně kopírují logiku pozitivních a negativních informací. Zde je ovšem potřeba dodat, že v pozitivním reportu lze najít jak negativní, tak pozitivní data o subjektu, ale dle zákonů platných v České republice, je k jejich přístupu potřeba získat souhlas daného subjektu.

### **3.2.1 Centrální Registr Úvěrů (CRÚ)**

Na přelomu tisíciletí vykazoval bankovní sektor nestabilitu, která byla způsobena nedostatečným sdílením informací, což vedlo k velkému kreditnímu riziku všech bank na trhu. Právě to, spolu se zkušenostmi z ostatních států EU, vedlo ČNB ke kroku vytvořit registr s jednoduchým cílem. Posílit stabilitu bankovního sektoru v České republice. V první polovině roku 2000 proto rozhodla Česká národní banka (ČNB) o vzniku a následném provozování Centrálního registru úvěrů (CRÚ). (Centrální registr úvěrů, 2021)

Centrální registr úvěrů byl uveden do ostrého provozu v roce 2002, poté co splnil všechny legislativní požadavky plynoucí z dodatků Zákona o bankách, včetně ochrany osobních údajů. (Mejstřík, 2014, s.358)

Registru CRÚ se účastní všechny banky a pobočky zahraničních bank, které na území České republiky působí. Dále se jej mohou účastnit i další osoby, pokud tak stanoví zvláštní zákon. Pro banky je účast v registru povinná.

CRÚ shromažďuje informace o úvěrových závazcích fyzických osob podnikatelů a právnických osob, tedy v podstatě o subjektech, které mají přidělené IČO, a posléze umožňuje výměnu těchto dat mezi svými uživateli. Přístup k těmto informacím má stejná skupina subjektů, která se registru účastní, a navíc ještě Česká národní banka jako provozovatel registru. Data jsou aktualizována na měsíční bázi.

Uživatel registru má tak přístup k současným závazkům klienta a k až desetileté historii.

Dle dat z roku 2013 bylo v CRÚ vedeno v registru přes 611 tisíc dlužníků, registr tou dobou užívalo více než 2 800 subjektů a počet dotazů do registru se pohyboval kolem 3,8 milionu za rok.

ČNB se později, díky CRÚ, zapojila i do programu výměny úvěrových dat mezi úvěrovými registry dalších států EU. Dochází tak k výměně například se Španělskem, Francií, Itálií, Rakouskem a Německem. Touto cestou pak ČNB získává data o českých klientech, kteří jsou úvěrováni v zahraničí a naopak. (Mejstřík, 2014, s.360)

### **3.2.2 Registr Fyzických Osob (Registr FO)**

Registr Fyzických Osob je jedním ze tří úvěrových registru pod záštitou Sdružením na Ochranu Leasingu a Úvěrů Spotřebitelům. Jsou do něj shromažďována pouze negativní data o klientech uživatelů registru, kterými jsou fyzické osoby. (SOLUS - Zájmové sdružení právnických osob, 2021)

Jedná se tedy pouze o data o klientech, kteří nejsou ochotní nebo nejeví zájem plnit své závazky nebo se dostali do finančních problémů, kvůli kterým nemohou svých závazků dostát.

Registr Fyzických Osob je provozován na základě Zákona o ochraně spotřebitele, ve znění pozdějších předpisů, respektive podle paragrafů § 20z a § 20za.

Přístup k datům je možný na základě reciproční výměny. Tedy dotazovat se do registru mohou pouze jeho aktivní přispěvatelé. Data jsou v Registru fyzických osob aktualizována na měsíční bázi. Díky tomu mají uživatelé možnost mít vždy k dispozici sadu aktuálních dat.

Negativní záznam obsahuje následující data:

- Jeden z užívaných negativních příznaků
  - A – klient dluží první dvě splátky
  - B – klient dluží tři splátky
  - C – klient dluží poslední splátku
  - D – z důvodu nesplacení dlužné částky došlo k zesplatnění pohledávky
  - P – pohledávka byla odprodána
  - U – dluh po splatnosti již není věřitelem vymáhán nebo nemá informace o stavu vymáhání nebo byla pohledávka ukončena, popřípadě odepsána
  - Z – došlo k zaplacení pohledávky
- datum, ke kterému negativní příznak vznikl,
- produkt, o který se jedná,
- dlužnou částku po splatnosti,
- počet dlužných splátek,
- datum, ke kterému byla pohledávka zaplacená, odprodána nebo odepsána.

Ke konci září 2020 bylo v negativním Registru fyzických osob sdružení SOLUS zapsáno 520 tisíc osob, které dlouhodobě neplní své splatné závazky. (SOLUS - Zájmové sdružení právnických osob, 2021)

### **3.2.3 Registr podnikatelů a právnických osob (Registr IČ)**

Druhým úvěrovým registrem řazeným pod Sdružení na Ochranu Leasingu a Úvěrů Spotřebitelům je Registr Podnikatelů a Právnických Osob. Ze zkratky názvu, Registr IČ, lze vyčíst, že v tomto registru jsou sdružována data o subjektech, které mají přidělené své Identifikační číslo osoby (IČO), tedy o fyzických osobách podnikatelích a o právnických osobách.

Podobně jako v případě Registru Fyzických Osob, jsou v registru vedena opět pouze negativní data o subjektech, klientech uživatelů registru, kteří nejsou ochotni nebo schopni svým závazkům dostát. (SOLUS - Zájmové sdružení právnických osob, 2021)

Provoz Registru podnikatelů a právnických osob je umožněn na základě Zákona o ochraně spotřebitele, ve znění pozdějších předpisů, respektive podle paragrafů § 20z a § 20za.



Informace jsou aktualizovány každý měsíc a jsou dostupné aktivním přispěvatelům registru na základě pravidla reciprocity. Výstup z registru pak může uživateli pomoci v rozhodování ohledně nabídky nové služby pro klienta. Výstup může také pomoci určit jak, a případně zda vůbec, přistupovat k vymáhání pohledávek.

V Registru podnikatelů a právnických osob je užitá obdobná logika obsahu negativní informace jako v případě Registru Fyzických Osob. Lze v něm tedy nalézt následující data:

- jeden z užívaných negativních příznaků
  - A – klient dluží první dvě splátky,
  - B – klient dluží tři splátky,
  - C – klient dluží poslední splátku,
  - D – z důvodu nesplacení dlužné částky došlo k zesplatnění pohledávky,
  - P – pohledávka byla odprodána,
  - U – dluh po splatnosti již není věřitelem vymáhán nebo nemá informace o stavu vymáhání nebo byla pohledávka ukončena, popřípadě odepsána,
  - Z – došlo k zaplacení pohledávky,
- datum, ke kterému negativní příznak vznikl,
- produkt, o který se jedná,
- dlužnou částku po splatnosti,
- počet dlužných splátek,
- datum, ke kterému byla pohledávka zaplacená, odprodána nebo odepsána.

Počet firem a podnikatelů vedených v Registru IČ byl 47 tisíc. Takto nízká hodnota, i v porovnání s Registrem Fyzických Osob, je způsobena dvěma skutečnostmi. Zaprvé, v České republice je samozřejmě vyšší počet fyzických osob než osob disponujících Identifikačním číslem osoby. Zadruhé, registr opět obsahuje pouze negativní informace, což jeho naplněnost ještě více snižuje. (SOLUS - Zájmové sdružení právnických osob, 2021)

### **3.2.4 Pozitivní registr**

Třetím, a zatím posledním, úvěrovým registrem provozovaným Sdružením na Ochranu Leasingu a Úvěrů Spotřebitelům je Pozitivní registr.

Pozitivní registr shromažďuje a zpracovává údaje o klientech a jejich smluvních závazcích. (SOLUS - Zájmové sdružení právnických osob, 2021)

Pozitivní registr, navzdory svému názvu, nezpracovává pouze pozitivní informace. Data, která jsou Pozitivním registrem zpracovávána jsou jak pozitivního, tak negativního charakteru. To znamená, že nehovoří pouze o závazcích, které klienti uživatelů Pozitivního registru nejsou ochotni nebo schopni uhradit, ale i o závazcích, které jsou řádně plněny.

Tato skutečnost umožňuje uživatelům Pozitivního registru daleko lépe, přesněji a v širší míře vyhodnotit, zda je jejich klient schopen unést další finanční závazek, případně zda je schopen dostat těm stávajícím. (SOLUS - Zájmové sdružení právnických osob, 2021)

Výhodou na klientské straně je pak skutečnost, že na základě pozitivních informací o jeho stávajících závazcích, má klient větší šanci získat další službu od jiného uživatele Pozitivního registru, se kterým do té doby uživatel nejednal.

Pozitivní registr Sdružení na Ochranu Leasingu a Úvěrů Spotřebitelům je provozován na základě Zákona o ochraně spotřebitele, ve znění pozdějších předpisů, respektive podle paragrafů § 20z a § 20za. (SOLUS - Zájmové sdružení právnických osob, 2021)

### **3.2.5 Bankovní Registr Klientských Informací (BRKI)**

Bankovní registr klientských informací je provozován společností CBCB – Czech Banking Credit Bureau, a. s., která byla založen v roce 2002, za účelem vzniku registru. Vlastníky společnosti je pět zakládajících bank: Česká Spořitelna, Komerční Banka, MONETA Money Bank, Československá obchodní banka a UniCredit Bank. (CRIF - Czech Credit Bureau, 2021)

BRKI, podobně jako CRÚ, se od ostatních registrů liší tím, že je založen na základě Zákona o bankách. Ten ukládá bankám povinnost sdílet data o svých klientech a jejich závazcích z důvodu předcházení zadlužování klienta nebo vzniku podvodů ze strany klienta.

Databáze obsahuje pozitivní i negativní data fyzických osob a fyzických osob podnikatelů, která vypovídají o bonitě a důvěryhodnosti klientů uživatelů registru. Data jsou aktualizována na měsíční bázi. Jsou přístupná na bázi reciprocity, pro uživatele registru, po celou dobu trvání závazku a 4 roky po jeho ukončení. V případě žádostí o úvěr je to 1 rok. (CBCB.cz - Bankovní registr klientských informací, 2021)

O klientech uživatelů jsou v registru vedeny následující informace:

- osobní a identifikační údaje v případě fyzické osoby
  - jméno,
  - příjmení,
  - rodné číslo,
  - datum narození,
  - místo narození,
  - údaje o adresách,
  - kontaktní údaje, tedy telefonní čísla, případně email,
  - dokumenty (občanský průkaz, cestovní pas a podobně)
    - sériové číslo,
    - datum vystavení,
    - datum platnosti,
- osobní a identifikační údaje v případě fyzické osoby podnikatele a právnické osoby
  - obchodní název podnikatele nebo společnosti,
  - identifikační číslo osoby,
  - právní forma subjektu,
  - údaje o adrese.

Uživatelem registru se mohou stát pouze banky nebo pobočky zahraničních bank, které mají s CBCB uzavřenou tzv. Smlouvu o zpracování údajů. Dále registr BRKI sdílí data s registrem NRKI a jeho uživateli, a to od roku 2006. Výměna nemůže být provedena bez souhlasu, které vybírají oba registry. Následné výstupy z registrů mají společně vytvořit ucelený obraz o závazcích klienta.

### **3.2.6 Nebankovní Registr Klientských Informací (NRKI)**

Nebankovní registr klientských informací je provozován zájmovým sdružením právnických osob Czech Non-Banking Credit Bureau, z.s.p.o., které bylo založeno v roce 2004. Za zakládající členy je bráno sedm společností z oblasti leasingu a splátkového prodeje. Jsou jimi CAC LEASING, a.s., CCB Finance, a.s., ČSOB Leasing, a.s., GE Money Multiservis, a.s., GE Money Auto, a.s., Leasing České spořitelny, a.s., a ŠkoFIN s.r.o. Samotný registr existuje na základě zákona o Ochráně spotřebitele. (CRIF - Czech Credit Bureau, 2021)

Účelem tohoto registru je získávat a sdílet data o bonitě, důvěryhodnosti a platební morálce osob (FO, FOP i PO), které jsou klienty uživatelů registrů. Obsahem registru jsou pozitivní i negativní informace o klientech zúčastněných finančních institucí.

Datová výměna probíhá, podobně jako u jiných registrů, na bázi reciprocity. K aktualizaci dat dochází měsíčně.

Registr uchovává data, po celou dobu trvání závazku, 4 roky po ukončení závazku a 6 měsíců v případě žádostí o úvěr. Uživatelům registru je tak umožněno získat ucelenější obraz o úvěrové historii klienta, na základě které může dále rozhodnout o vyhovění jeho žádosti. (CNCB - Nebankovní registr klientských informací, 2021)

Nebankovní Registr Klientských Informací vede data o klientech svých uživatelů, podobně jako Bankovní Registr Klientských Informací, v následující logice:

- osobní a identifikační údaje v případě fyzické osoby
  - jméno,
  - příjmení,
  - rodné číslo,
  - datum narození,
  - místo narození,
  - údaje o adresách,
  - kontaktní údaje, tedy telefonní čísla, případně email,
  - dokumenty (občanský průkaz, cestovní pas a podobně)
    - sériové číslo,
    - datum vystavení,
    - datum platnosti,
- osobní a identifikační údaje v případě fyzické osoby podnikatele a právnické osoby
  - obchodní název podnikatele nebo společnosti,
  - identifikační číslo osoby,
  - právní forma subjektu,
  - údaje o adrese,

Uživateli registru, kteří jsou v rámci něj sdružováni jsou leasingové a úvěrové společnosti, kterých je v registru cca čtyřicet. Uživatelem registru se může stát, stejně jako u BRKI, pouze takový subjekt, který má s CNCB uzavřenou tzv. Smlouvu o zpracování údajů.

Od roku 2006 dochází k vzájemné výměně dat mezi NRKI a BRKI, který je provozován již dříve zmiňovanou společností CBCB – Czech Banking Credit Bureau, a. s. Oba registry ovšem musejí vybírat souhlasy ke sdílení, neboť bez nich není výměna dat možná. To umožňuje, v případě, kdy jsou data určena ke sdílení mezi registry, získat finančním institucím informace o klientově kompletní úvěrové angažovanosti.

### **3.2.7 Registr Platebních Informací (REPI)**

Registr REPI je zatím nejmladším přírůstkem do seznamu registrů provozovaných v ČR. Je provozován společností CRIF – Registr platebních informací s.r.o.. REPI jako takový byl založen v roce 2017. Podobně jako ostatní registry je provozován na základě zákona o ochraně spotřebitele. REPI pod sebou sdružuje společnosti, které svým klientům (spotřebitelům) poskytují služby, které jsou založené na formě splácení. (CRIF - Czech Credit Bureau, 2021)

Registr je určen pro poskytovatele úvěrů e-shopy, společnosti pronajímající nemovitosti, poskytující telekomunikační nebo datové služby nebo poskytovatelé kabelových televizí. (CRIF - Czech Credit Bureau, 2021)

V zásadě se tedy jedná o subjekty ze specifických odvětví, ve kterých se, ve velké míře, data nesdílela.

Registr funguje na principu reciprocity, tedy pokud do registru uživatel nepřispívá, nemůže se do něj dotazovat. To samé platí i o kvalitě dat. Tedy pokud sdílí jen negativní informace o svých spotřebitelích, může se dotazovat pouze na negativní data o spotřebitelích ostatních přispěvatelů.

REPI zpracovává pozitivní i negativní data, a to o fyzických osobách, fyzických osobách podnikatelích i právnických osobách. (CRIF - Czech Credit Bureau, 2021)

Dále, díky tomu, že je registr REPI multisegmentovým registrem (jsou v něm sdružena data od přispěvatelů z více odvětví trhu), umožňuje takzvané segmentové dotazy. Tedy dotazy, jejichž odpovědí jsou data pouze za odvětví, ve kterém se uživatel pohybuje. I zde je užito principu reciprocity ve smyslu kvality dat.

Aktualizace dat v tomto registru probíhají na pravidelné bázi, vždy po dohodě s uživatelem. Díky specifitě odvětví mikropůjček se jedná o rozpětí od jednoho dne až po jeden měsíc.

Registr REPI operuje s daty o 400 000 unikátních subjektech asi 621 000 unikátních platebních záznamech od zhruba třicet přispěvatelů. (Kučera, 2020, s.35)

### **3.3 Právní rámec a zpracování osobních údajů**

Nový občanský zákoník nabízí definici spotřebitele jako „každého člověka, který mimo rámec své podnikatelské činnosti nebo mimo rámec samostatného výkonu svého povolání uzavírá smlouvu s podnikatelem nebo s ním jinak jedná“. (Ondřej, 2013, s.13)

#### **3.3.1 Zákon o ochraně spotřebitele**

Existenci většiny zmíněných registrů, krom BRKI, umožňuje Zákon č. 634/1992 Sb., Zákon o ochraně spotřebitele, respektive Část pátá - Informační databáze o bonitě a důvěryhodnosti spotřebitele.

Podmínkou pro vznik registru je účast alespoň 10 prodávajících, kteří splňují zákonnou definici banky, poskytovatele spotřebitelského úvěru, platební instituce nebo poskytovatele platebních služeb malého rozsahu.

Údaje získané podle odstavce 1 jsou prodávající oprávněni použít výhradně k předcházení podvodnému jednání a k posouzení schopnosti a ochoty spotřebitele splnit jeho smluvní závazky (posouzení bonity, platební morálky a důvěryhodnosti). (Sbírka zákonů České republiky)

To v podstatě znamená to, že uživatelé registru mohou data, která získají z registru následně užít jak k ochraně svých vlastních zájmů, tak k ochraně spotřebitele, který se díky tomu nemusí dostat do finančních problémů.

Na základě dat, která jsou do registru nahrána je možné, samozřejmě při relevantním množství, vytvářet modely. Ty mohou vyhodnocovat pravděpodobnost podvodného jednání nebo míru schopnosti a ochoty spotřebitele plnit své závazky vůči prodávajícím, respektive uživatelům registru.

Prodávající, který poskytuje údaje spotřebitele do registru, je povinen spotřebitele předem upozornit na skutečnost, že jeho údaje budou nebo mohou být vedeny v registru a v případě, že údaje obsahují informace o dluhu spotřebitele po splatnosti, je dále povinen informovat spotřebitele o existenci dluhu a vyzvat jej k úhradě takového dluhu. (Sbírka zákonů České republiky)

To znamená, že prodávající nemůže bez vědomí spotřebitele reportovat data o něm do registru.

V registru lze zpracovávat následující údaje

a) identifikační údaje spotřebitele,

- b) údaje o finančních závazcích spotřebitele ze smluv mezi spotřebitelem a prodávajícími uživateli registr, včetně údajů o potenciálních finančních závazcích spotřebitele ze smluv, o kterých spotřebitel s prodávajícím jednal, ale k jejichž uzavření nedošlo,
- c) časové určení období, ke kterému se údaje vztahují,
- d) údaj o prodávajícím, který záznam zapsal.

(Sbírka zákonů České republiky)

ad a. Mezi identifikační údaje, které může provozovatel registru zpracovávat patří například i rodné číslo. To je všeobecně bráno jako univerzální identifikátor. Pravidlo neplatí vždy, neboť například Slovenská republika, i po rozdělení Československa v roce 1993, užívá stejnou logiku rodného čísla jako Česká republika. Proto je nutné k rodnému číslu přiřadit další informaci jako například stát, ve kterém se jeho držitel narodil, aby se tím předešlo možné záměně při vyhledávání spotřebitele nebo ztrátě informací v případě spojování dat od více uživatelů. Dalšími identifikačními údaji jsou jméno, příjmení, datum narození, případně adresa trvalého bydliště, která ovšem může být zavádějící hlavně v případech, kdy například rodič a potomek, se stejným křestním jménem, žijí na stejné adrese.

ad b. Ani tyto údaje nemůže registr držet neomezeně dlouho. Data o závazcích spotřebitele zpracovávat provozovatel registru po dobu trvání závazku a následně 3 let po jeho zániku. Závazek může zaniknout jeho splacením, promlčením nebo osvobozením spotřebitele od plnění závazku. V případě projednávaných ale neuzavřených závazků může registr data zpracovávat pouze po dobu 3 měsíců. Po uplynutí vymezené doby, ať 3 let nebo 3 měsíců, je prodávající nebo provozovatel registru oprávněn údaje dále uchovávat za statistickými účely, ale již v anonymizované formě.

V případě, kdy se uživatel dotazuje na spotřebitele do registru, který ve svém výstupu obsahuje i pozitivní data, tedy závazky, u kterých nedošlo k prodlení, je uživatel povinen získat od spotřebitele souhlas k takové akci. V praxi to znamená, že v situaci, kdy například za poskytovatelem úvěrů přijde spotřebitel s úmyslem požádat o úvěr, má poskytovatel možnost nahlédnout do registru, zda, a případně jaké informace o spotřebitele registr vede, ovšem pouze se souhlasem spotřebitele, bude-li se jednat i o pozitivní data.

Provozovatel registru je dále povinen uzavřít s uživatelem registru písemnou smlouvu, ve které je smluvně zaváže k dodržování všech zákonem požadovaných povinností. Smlouva dále musí obsahovat i ujednání o organizačně-technických opatřeních k zabezpečení údajů.

Vznik registru prodávající nebo provozovatel, který jej vede, oznámí Úřadu pro ochranu osobních údajů podle jiného zákona upravujícího ochranu osobních údajů. (Sbírka zákonů České republiky)

Zákon o ochraně spotřebitele ovšem obsahuje i část, ve které se hovoří o přístupu spotřebitele k datům, která jsou o něm vedena, a o ochraně spotřebitele v případech, kdy jsou o něm v registrech vedeny nepravdivé nebo nepřesné informace.

V první řadě má spotřebitel právo, za úhradu odůvodněných nákladů, získat výpis z registru, který obsahuje všechna data, která jsou o něm v registru vedena.

Dále má právo, v případě, že data o něm vedená jsou nepřesná nebo nepravdivá, požadovat po provozovateli nebo prodávajícím nápravu. Provozovatel je povinen tyto informace bez prodlení opravit. V situaci, kdy nejsou data opravena bezprostředně, je provozovatel nebo prodávající povinen rozporovaná data, až do vyřešení situace, blokovat. Místo údajů poskytne provozovatel nebo prodávající informaci o tom, že jsou daná data blokována.

V případě, kdy jsou o něm v registru vedena data o závazcích, u kterých nedošlo k prodlení nebo se jedná o potenciální závazky, tedy o závazky, které nebyly uzavřeny, má spotřebitel právo požadovat po provozovateli nebo po prodávajícím výmaz takových dat. Provozovatel je poté povinen tyto informace bez prodlení vymazat a zajistit, aby pozitivní data o daném spotřebiteli již nebyla v registru vedena.

Pokud jsou údaje vedené v registru předmětem sporu mezi příslušným prodávajícím a spotřebitelem, přičemž o takovém sporu je vedeno soudní, správní nebo rozhodčí řízení, v němž nebylo doposud vydáno pravomocné rozhodnutí ve věci samé, nebo mimosoudní řešení spotřebitelského sporu podle tohoto zákona, je prodávající nebo provozovatel, který registr vede, povinen tuto skutečnost na žádost spotřebitele v registru poznamenat. (Sbírka zákonů České republiky)

### **3.3.2 Zákon o bankách**

Výjimkou je registr BRKI, který je založen na základě Zákona č. 21/1992 Sb., Zákona o bankách.

V rámci plnění povinnosti postupovat při výkonu své činnosti obezřetně se banky a pobočky zahraničních bank mohou vzájemně informovat o bankovním spojení, identifikačních údajích o majitelích účtů a o záležitostech, které vypovídají o bonitě a důvěryhodnosti jejich klientů, a to i prostřednictvím právnické osoby, která není bankou.



Majetkový podíl na této právnické osobě mohou mít pouze banky, které jsou povinny zajistit, že tato právnická osoba bude zachovávat získané údaje v tajnosti a chránit je před zneužitím. Banka a pobočka zahraniční banky je povinna k získaným údajům o klientech jiné banky a pobočky zahraniční banky přistupovat tak, jako by šlo o údaje o jejích vlastních klientech. (Sbírka zákonů České republiky)

Z praxe je pak ještě nutné dodat, že pro provozovatel registru je vždy lepší, pokud má k dispozici takzvaný Balanční test.

### **3.4 Databáze**

Nejprve je třeba vysvětlit, co se skrývá pod pojmem registr. Na registr lze pohlížet jako na složitou/sofistikovanou databázi. Databáze je organizovaný soubor strukturovaných informací neboli dat, které se obvykle ukládají v elektronické podobě v počítačovém systému. (Oracle, 2021)

#### **3.4.1 Historie**

Potřebu shromažďovat a ukládat data, respektive informace, mělo lidstvo již od svého vzniku. Ať se jednalo o množství zásob, počty zvířat, nebo později i například seznamy dlužníků. Na tyto seznamy lze pohlížet jako na primitivní či méně primitivní předchůdce databází.

O prvních pokusech o databáze lze hovořit až s příchodem dřevěných štítků a papírových pásek, které ovšem nedokázaly splnit požadavky na zpracování velkých množství dat. Jako příklad prvního užití dřevěných štítků je uváděno sčítání lidu v roce 1890 v USA. Zpracování takto nashromážděných dat pak probíhalo na elektromechanických strojích.

Jako přelom se všeobecně bere nástup magnetických záznamových médií. Vynález magnetického záznamového média, jako jsou magnetické pásky a magnetické disky, umožnil trvalé uložení „velkého“ množství dat, které již umožňovalo počítačové zpracování. (Tyrychtr, 2015, s.8)

Magnetická média jsou využívána i v dnešní době, a to například jako pevná záloha elektronických databází pro případ, kdy by došlo k problému s infrastrukturou případně k jejímu napadení.

## 3.5 Databázové modely

Databáze lze rozdělit do několika základních typů, a to na základě způsobu ukládání dat a vazeb mezi nimi.

### 3.5.1 Hierarchické databáze

Vznik hierarchického modelu dat se datuje do sedmdesátých let minulého století, kdy byl vyvinut společností International Business Machines Corporation (IBM). Důvodem k vývoji tohoto modelu byla nulová flexibilita magnetických disků vzhledem k postupnému přístupu k datům.

Data jsou v tomto modelu organizována v takzvané stromové struktuře. To znamená, že každý záznam funguje jako uzel.

Zmíněný model poté umožnil, aby hierarchické záznamy, a to jakékoliv, mohly být zapsány jako posloupnost. Následně byla pak takováto sekvence uložena na magnetické pásky.

Původní použití určené pro IMS bylo „účtování zpracovaného materiálu“ a vybraný model byl pro tento účel ideální. (Tyrychtr, 2015, s.9)

Určitou nevýhodou hierarchického modelu dat byla neschopnost zachytit sémantiku dat, která byla na magnetickém pásku uložena. Model dokázal pouze reprezentovat vztahy mezi daty.

Jednotlivá pole nebyla identifikována v systému pro správu databází, záznam byl definován jednoduše jako počet bytů, do kterých mohou být data umístěna. (Tyrychtr, 2015, s.9)

Popisovaná nedostatečnost modelu způsobovala to, že databáze nepodporovala takzvané ad hoc dotazy.

Pro přístup do databáze bylo nutné psát nové programy. Jediné, co bylo vloženo do programů, které aplikace vytvořila, bylo zpracování sémantiky.

Nedostatky hierarchického modelu vedou organizace, které stále ještě IMS užívají, k implementaci datových skladů. Mezi subjekty, které stále IMS využívají patří například bankovní společnosti, a to zejména v případě aplikací, jichž vývoj je stabilní a pomalý.

### 3.5.2 Síťové databáze

Nástupcem a částečnou nápravou hierarchického modelu dat se stal síťový model dat. Ten odstranil omezení ve vztazích, jejichž zastoupení v hierarchickém modelu nebylo možné.

Hlavním problémem vztahů Hierarchického modelu byla skutečnost, že v něm šlo realizovat pouze vztahy jeden k mnoha. V konečném důsledku to znamená, že jeden uzel může mít pouze jeden rodičovský uzel. Tento nedostatek byl v Síťovém modelu dat doplněn, a díky tomu mohl jeden uzel souviset s více dalšími uzly než jedním.

### 3.5.3 Relační databáze

Současníkem již zmíněného Síťového modelu dat byl právě Relační model dat. Síťový model dat se ovšem dokázal rozšířit do komerčních databázových systémů daleko rychle, než relační model dat. (Tyrychtr, 2015, s.12)

Důvodem, proč se v rozšiřování relačního modelu dat pokračovalo bylo několik postřehů, se kterými přišla IBM.

Bylo nutné zvýšit nezávislost dat, která byla v systémech pro správu dat uchovávána. Stávající Síťové a hierarchické systémy dat totiž trpěly závislostí „přístupové cesty“ k datům.

Tato závislost znamenala to, že návrhář, který databázi připravoval musel umět předvídat, které aplikace a jak budou s databází následně provázány. V momentě, kdy se objevila nová aplikace, která nebyla podporována, muselo dojít k přepracování celé databáze. Snahou relačního modelu dat bylo naprosto se vyhnout předdefinovaným přístupovým cestám do databáze. (Tyrychtr, 2015, s.12)

Bylo potřeba změnit přístup k ukládání a následnému načítání dat. Zvoleným přístupem se měl stát přístup matematický.

A zatřetí, modely dat musely začít podporovat zpracovávání ad hoc dotazů.

K první rozsáhlejší implementaci modelu došlo v roce 1976 ve výzkumném projektu s názvem System R, který byl zaštiťován právě společností IBM. V roce 1979 pak vznikla první komerční relační databáze, Oracle, která v sobě nesla mnoho poznatků, zjištění a myšlenek získaných v rámci projektu System R. (Tyrychtr, 2015, s.13)

Dominantními se databáze založené na relačním modelu dat staly v osmdesátých letech. Relační model dat, díky své vysoké efektivitě a flexibilitě, začal, postupem času,

získávat na významu, a proto vznikla potřeba vytvořit standardizovaný dotazovací jazyk, který by byl v těchto databázích užíván. Tim se stal Structured Query Language (SQL), který byl přímým nástupcem jazyka SEQUEL. Ten měl za cíl se co nejvíce přiblížit syntaxi přirozeného jazyka, tehdy angličtině a byl vyvíjen společně s projektem System R.

#### **3.5.4 Objektové databáze**

Objektové modely dat nebo Objektově orientované databáze jsou jednou z podskupin takzvaných NoSQL databází, které nejsou relačními databázemi. Objektové databáze v posledních letech zaznamenávají značný vzrůst zájmu. Zkratka NoSQL značí, že databáze užívá jiných prostředků k ukládání a zpracování dat, než standardní jazyk SQL. To ovšem neznamená, že je databáze provozována čistě bez SQL, naopak mnoho databází umožňuje dotazy v SQL a bere v potaz obě možnosti přístupu.

Objektové databáze jsou úzce spjaté s objektovým programováním. V tomto druhu programování je objekt brán jako entita zájmu v dané aplikaci. (Oracle, 2021)

Objekt má stav popisující specifické strukturální vlastnosti objektu. Objekt má také chování, definuje metody, které se používají k manipulaci s objekty. (Tyrychtr, 2015, s.15)

O prvních objektových databázových systémech lze hovořit již od první poloviny osmdesátých let. Na jejich normalizaci se pracuje již od roku 1993.

V současnosti lze uvést jako typický příklad využití Objektové databáze Geografický informační systém (GIS). (Tyrychtr, 2015, s.15)

### **3.6 Databázové systémy**

Dle definice společnosti Oracle je databáze organizovaný soubor strukturovaných informací neboli dat, které se obvykle ukládají v elektronické podobě v počítačovém systému. Samotná databáze je pak řízena takzvaným DBMS, což je systém pro správu databáze. Ve spojení s dalšími aplikacemi je pak celek označován jako databázový systém. Toto slovní spojení je pak běžně zkracováno na výraz databáze. (Oracle, 2021)

V současné době, se již z důvodu úspory jak financí, tak času, nejeví jako rozumné, aby si společnost vytvářela databázový systém sama. Společnosti pak často volí řešení ve formě připravených databázových systémů. Výhodou outsourcingu, vyčlenění určitých činností směrem k externímu dodavateli, je pak stálá podpora, pravidelná aktualizace systému ze strany dodavatele a v neposlední řadě mnohdy nižší finanční náročnost. Mezi dodavatele takovýchto databází pak patří například již zmiňovaná společnost Oracle.

Podobně jako v případě Frameworků, lze získat i databázové systémy v open source licenci. Je třeba doplnit, že hlavní předností open source není jeho bezplatnost, ale skutečnost, že není vyvíjen jednou společností, ale různými nezávislými programátory. To zajišťuje více rozličných pohledů na danou problematiku. V momentě, kdy je program shledán dostatečně funkčním, je programátory ohlášena jeho dostupnost. Tomuto přístupu k vývoji programů značně přispělo rozšíření Internetu mezi uživateli. Mezi takové se řadí například PostgreSQL.

### 3.6.1 PostgreSQL

Jedná se o nejvyspělejší databázový systém spadající do skupiny Open Source software. Postgres původně vznikl na kalifornské univerzitě v Berkley. Jednalo se o objektově relační databázový server. Jeho předchůdcem byl systém Ingres, který byl vyvíjen tamtéž, a to v letech 1977 až 1985. Zdrojový kód Postgres byl následně převzat společností Illustra. Zdrojový kód systému je psán v jazyce C a obsahuje 250 tisíc řádků. Ten byl následně dvěma studenty univerzity v Berkley, Jollym Chenem a Andrewem Yuem, doplněn o podporu jazyka SQL. Tím vznikl Postgres95, který Chen, po opuštění univerzity, nadále udržoval. (Momjian, 2003, s.1)

Následně v roce 1996, kdy byla rozpoznána potřeba Open Source SQL databázového serveru, byla sestavena skupina pracovníků, kteří se začali zabývat dalším vývojem systému Postgres95. Na konci téhož roku došlo k přejmenování systému z Postgres95 na dnešní PostgreSQL.

Prvním cílem skupiny, která na svém počátku čítala pouze čtyři členy, kteří navíc na systému pracovali ve svém volném čase, bylo utřídit staré příspěvky a vyhodnotit patche. Patche jsou části zdrojového kódu, které doplňují funkce nebo opravují chyby softwaru. Po přetvoření původního univerzitního kódu pak bylo možné v pravidelných intervalech, zpravidla 3-5 měsíců, vydávat nové verze systému.

Po úspěšném startu byl systém začleněn do distribuce Linux Red Hat, což napomohlo jeho rychlejšímu rozšíření. (Momjian, 2003, s.4)

PostgreSQL je šířen pod BSD licenci, která je nejliberálnější ze všech open source licencí. Dále Štědroň zdůrazňuje, že předností systému PostgreSQL je rozšiřitelnost. Tedy schopnost systému být rozšiřován o nové funkce, datové typy nebo procedurální jazyky. (Štědroň, 2009, s.217)

### 3.6.2 MySQL

Dalším databázovým systémem je MySQL, který vytvořila švédská firma MySQL AB. MySQL je nejoblíbenější databázový systém s veřejným zdrojovým kódem. Jedná se o systém určený pro relační databáze. (Prokopová, 2006, s.28)

Díky jeho dvojímu licencování, bezplatné licenci GPL a komerční placené licenci, je považován za průkopníka v dané oblasti. Podobně jako PostgreSQL komunikuje i MySQL za pomoci jazyka SQL.

Hlavními uživateli tohoto systému jsou Facebook, Twitter, Youtube, Yahoo!, Flickr nebo Wikipedia.

V současné době je provozována společností Oracle, která ji získala odkupem společnosti Sun Microsystems v roce 2010.

Zároveň je důležitou součástí softwarového balíčku LAMP, který je vývojářskou platformou. Ten sdružuje Linux, jako operační systém, Apache jako webový server, MySQL v roli databáze a jako skriptovací jazyk používá PHP, ten může být zaměněn za Python nebo Perl.

Mezi hlavní výhody daného databázového systému patří jeho snadná implementovatelnost, schopnost volného šíření a výkon. Díky těmto skutečnostem má vysoký podíl na databázích v současné době používaných. (Štědroň, 2009, s.216)

Díky podobnosti jazyka SQL a jazyka platformy MySQL, se uživatel nemusí učit žádný nový jazyk, ale může používat standardní SQL příkazy.

Je nutné dodat, že od začátku se MySQL zaměřoval primárně na rychlost, a to na úkor některých zjednodušení, například v oblasti způsoby zálohování.

### 3.6.3 Oracle Database

Oracle je systém řízení báze dat (Oracle database management system – DBMS), moderní multiplatformní databázový systém s velice pokročilými možnostmi zpracování dat, vysokým výkonem a snadnou škálovatelností. V České republice operuje společnost od roku 1994. (Procházka, 2009, s.33)

Na rozdíl od dříve zmíněných systémů je databáze od Oracle první plně placenou databází a zároveň není distribuována pod licenci open source. Společnost Oracle zákazníkům svůj software pronajímá. Software se tedy stává určitou formou služby a jeho uživateli pak odpadá starost o administrativní či správní náklady.

Jedná se o relační databázový systém. Ten podporuje několik programovacích jazyků. A to dotazovací jazyk SQL, firemní rozšíření Oracle, programovací jazyk PL/SQL (rozšiřuje možnosti samotného SQL). Dále systém podporuje objektové databáze a databáze uložené v hierarchickém modelu. (Procházka, 2009, s.35)

Nejnovější iterací systému Oracle je verze 21c, která byla, v cloudové verzi, nasazena v prosinci roku 2020. Česká pobočka společnosti Oracle pak nabízí verzi 19c, a to jak v cloudovém, tak v on premise řešení.

Databáze Oracle je podporována většinou hlavních systémových platform. Jmenovitě ji lze provozovat na Windows, UNIX, Linux nebo Mac OS.

#### **3.6.4 Microsoft SQL Server**

Jedná se o systém, jehož vznik je datován do přelomu osmdesátých a devadesátých let minulého století. Je distribuován jako open source. Od roku 1994 byl, po ukončení spolupráce mezi společnostmi Microsoft a Sybase, které byly původními tvůrci, vyvíjen pouze společností Microsoft. Microsoft SQL Server je relační databází a spolu s databázemi od společností Oracle a IBM je považován za vedoucí databázový systém na současném trhu. (Hughes, 2019)

Mezi roky 1995 a 2016 vypustila společnost Microsoft 10 verzí svého systému, přičemž poslední nasazenou verzí je SQL Server 2019 pod kódovým označením „Seattle“, která byla představena v listopadu 2019.

Na přelomu tisíciletí došlo k vypuštění verze SQL Server 2000, která tímto označením ctíla číslování dalších produktů platformy Windows. Díky své vysoké škálovatelnosti a spolehlivosti je dodnes považována za pokus zaútočit na konkurenční databázi Oracle v tehdejší verzi 9i. (Knight, 2004, s.5)

Od verze z roku 2016 přidal Microsoft nová rozšíření, která umožnila provádět operativní analýzu v reálném čase. Uvedená verze byla první, která se držela filozofie Mobile first, respektive Cloud first. Jedná se o způsob vývoje, ve kterém je upřednostněna jedna cílová platforma a následně je systém upravován pro ostatní platformy. Zde bylo upřednostněno Cloud řešení před On-premise.

Další důležitou součástí verze 2016 byly 3 ochranné prvky. Jmenovitě:

- Always Encrypted, ten umožňuje uživatele upravovat data aniž by je musel nejprve odšifrovat,

- Row-level security, který umožňuje kontrolu datových přístupů na úrovni řádků v tabulce,
- Dynamic data masking, který automaticky skrývá citlivá data před uživateli s nedostačujícím oprávněním.

(Hughes, 2019)

### 3.6.5 MongoDB

System, provozovaný společností MongoDB, Inc., která byla založena v roce 2007. Samotný databázový systém pak existuje od roku 2009. Dle informací z webových stránek má MongoDB více než 22 600 zákazníků. Databázi lze získat jak v komunitní, open source, tak i komerční verzi. Komerční verze pak existuje ještě ve vylepšené verzi, která vznikala ve spolupráci s IBM. Pro zajímavost, počet stažení databázové platformy přesáhl 130 milionů, z čehož více než milion byl stažen pod univerzitní registrací. Mezi její uživatele v současnosti patří subjekty jako MetLife, New York Times nebo fyzikální laboratoř CERN. (MongoDB, 2021)

MongoDB je dokumentově orientovaná NoSQL databáze, která je určena pro velkokapacitní ukládání. Na rozdíl od jiných databází, která pro svůj provoz užívají systém tabulek a řádků, používá MongoDB svůj vlastní systém dokumentů a kolekcí. Kolekce lze považovat za ekvivalent tabulek relačních databází, neboť pod sebou sdružují sady dokumentů a funkcí. Hlavní předností je tedy podpora dat, která jsou vedena v nestandardním tvaru. (Guru99.com, 2021)

Databáze podporuje takzvané Ad hoc dotazy. To znamená, že dotazy nemusí mít předdefinované schéma, a tedy je lze pokládat na velmi konkrétní úrovni.

Další výhodou MongoDB je replikace. V principu dochází k tomu, že existují dvě, totožné, verze databáze. Jedna figuruje jako primární databáze, ke které její klient přistupuje. Data z ní se klonují do sekundární databáze a v případě výpadku nebo problému s primární databází, je sekundární, záložní, schopná plně zastoupit tu primární.

Předností MongoDB, podobně jako i dalších NoSQL databází, je skutečnost, že nevyžadují předdefinovaná schémata. Spolu s tím podporuje i varietu programovacích jazyků, což umožňuje ponechat data v jejich nativním jazyce. (Botelho, 2020)

Hlavní funkcionalitou MongoDB je takzvaná horizontální škálovatelnost. Tedy schopnost přidávat nebo případně ubírat prvky v systému. Popisovaná možnost z něj činí zajímavé řešení pro společnosti, které operují s big data aplikacemi.



Nevýhodou MongoDB je využití pouze jednoho master node (hlavního uzlu), přes který musí projít všechna data. To omezuje rychlost zápisu do databáze s ohledem na kapacitu a vytížení uzlu. (Botelho, 2020)

### **3.7 Databázové Frameworky**

Framework je softwarová struktura. Slouží jako podpora a do jisté míry i zjednodušení vývoje dané aplikace. Z pravidla obsahuje již předpřipravené knihovny, například API, díky kterým mají posléze vývojáři aplikace zjednodušenou svoji práci a mohou se plně věnovat tvorbě samotné aplikace, a šetří tak čas, který by museli věnovat tvorbě daných knihoven. (Spring, 2021)

Další nespornou výhodou takovýchto frameworků je skutečnost, že jsou distribuovány pod open source licencí, jako je tomu v případě Spring Framework nebo Micronaut. Uživatel za takový obsah, nemusí platit a, jako v případě Spring Framework a Micronaut, má zároveň jistotu, že se jedná o obsah, který byl již ověřen jinými uživateli před ním. Podobně jako v případě databázových systémů se i databázové frameworky dostávají do takzvaného enterprise solution. Jinými slovy se stávají výchozím řešením pro větší společnosti.

#### **3.7.1 Spring framework**

Jedná se o jeden z nejrozšířenějších frameworků, který v současné době využívají miliony koncových uživatelů. Mezi subjekty, které spolupracují se Spring patří například čínská společnost Alibaba, která se zabývá převážně obchodováním, americké společnosti Amazon, Google a Microsoft. (Spring, 2021)

Klíčovým prvkem frameworku Spring je podpora infrastruktury na aplikační úrovni. Jinými slovy výhodou užití Spring je skutečnost, že se vývojáři aplikace mohou soustředit na část business logiky spjatou s aplikační úrovní, zatímco framework zajistí ostatní.

Další nespornou výhodou je snaha Spring o integraci odvětvových bezpečnostních standardů. Integrace standardů vede i větší firmy k tomu, aby přistoupily k užití takového řešení. (Spring, 2021)

Spring také do jisté míry těží ze své komunity, která obsahuje jak začínající programátory, tak profesionály z oboru. Rozsáhlá varieta uživatelů umožňuje, podobně jako v případě některých open source databázových systémů, různé pohledy na danou

problematiku. Komunita dále vytváří velké množství průvodců, návodů, videí a fór, v rámci kterých pořádá i schůzky. V neposlední řadě také nabízí formální tréninky a nabízí možnosti certifikace v dané problematice. (Spring, 2021)

### **3.7.2 Micronaut**

Micronaut je framework typu JVM (Java Virtual Machine), který pro vystavění service užívá programovacích jazyků Java, Kotlin nebo Groovy. Díky podstatě JVM, jedná se totiž o modul, který zpracovává pouze mezikód a tedy pracuje rychleji, se Micronaut stává zajímavým pro užití v oblastech Internetu věcí nebo Android aplikací.

V porovnání s frameworkem Spring byl Micronaut vytvořen s funkcí server-less nebo aby podporoval aplikace na platformě Android. V případě funkce server-less se vše odehrává v prostředí cloud a v něm poskytovatel prostředí přiděluje výpočetní kapacity na vyžádání, respektive dle potřeby. (ThoughtWorks, 2019)

### **3.7.3 Hibernate**

Hibernate je open source object relation mapping (ORM) nástroj, který poskytuje framework pro mapování objektově orientovaných doménových modelů na relační databáze pro webové aplikace. Distribuován je pod GNU Lesser General Public licenci. (TheServerSide, 2016)

Projekt Hibernate byl započat v roce 2001 a měl nabídnout lepší persistentní schopnosti než EJB2 (Enterprise Java Beans 2). Na počátku roku 2003 došlo k nasazení verze Hibernate2. V prosinci roku 2018 byla nasazena verze Hibernate ORM 5.4.0 Final.

Podstatou objektově relačního mapování je kontejnerizace objektů a jejich abstrakci. Abstrakce umožňuje adresovat, přistupovat a manipulovat objekty, bez toho, aby musel brát v potaz jejich souvislost k datovým zdrojům. (TheServerSide, 2016)

Hibernate mapuje třídy Java do databázových tabulek a datové typy Java do datových typů SQL.

Další předností Hibernate je redukce řádků kódu za pomoci udržování mapování objektových tabulek a výsledek vrací ve formě Java objektů.

### 3.8 Metoda TOPSIS

Pro určení ideálních technologií pro užití při tvorbě prototypu je užitá metoda TOPSIS. Tato metoda posuzuje varianty z hlediska jejich vzdálenosti od ideální a bazální varianty. (Brožová, 2003, s.36)

Její postup spočívá ve výpočtu následujících kroků:

- převod minimalizačního kritéria na maximalizační podle vztahu

$$y'_{ij} = -y_{ij},$$

- konstrukce normalizované kritériální matice  $R = (r_{ij})$  podle vzorce

$$r_{ij} = \frac{y'_{ij}}{\sqrt{\sum_{i=1}^p y'_{ij}{}^2}},$$

(Sloupce matice R jsou vektory jednotkové délky.)

- výpočet normalizované vážené kritériální matice  $W = (w_{ij})$  dle vztahu

$$w_{ij} = v_j r_{ij},$$

- určení ideální varianty  $h$  s ohodnocením  $(h_1, \dots, h_m)$  a bazální variantu  $d$  s ohodnocením  $(d_1, \dots, d_m)$  vzhledem k hodnotám matice W,
- výpočet vzdálenosti jednotlivých variant od ideální varianty

$$d_i^+ = \sqrt{\sum_{j=1}^k (w_{ij} - h_j)^2}$$

a od bazálních variant

$$d_i^- = \sqrt{\sum_{j=1}^k (w_{ij} - d_j)^2},$$

- výpočet relativního ukazatele vzdálenosti jednotlivých variant od bazální varianty podle vzorce

$$c_i = \frac{d_i^-}{d_i^+ + d_i^-},$$

(Hodnoty ukazatelů se pohybují mezi 0 a 1, kdy hodnotu 0 nabývá bazální a hodnotu 1 ideální varianta.)

- seřazení variant, sestupně, podle hodnot  $c_i$ . Požadovaný počet variant s nejvyššími hodnotami ukazatele jsou považovány za řešení.

(Brožová, 2003, s.36-37)

## 4 Vlastní práce

Jak již vyplývá z literární části bakalářské práce, v České republice je pouze sedm registrů, které se zabývají solventností občanů. Z toho dva pod sebou sdružují data ze segmentu mikropůjček respektive mikroúvěrů. Jsou jimi Nebankovní Registr Klientských Informací (NRKI) a Registr Platebních Informací (REPI), které jsou technicky spravovány společnostmi CRIF - Czech Credit Bureau, a. s. Tedy je relevantní záměr Registru platební morálky porovnávat právě s nimi, neboť jeho hlavními uživateli by měli být poskytovatelé nebankovních úvěrů. Oba výše zmíněné registry figurují na trhu již několik let.

### 4.1 Požadavky na prototyp

Nejprve je potřeba stanovit požadavky, které má registr splňovat.

Registr je určen pro uživatele přistupující z osobního počítače, tedy jej není nutné vyvíjet pro telefon, případně tablety.

V první verzi bude registr zpracovávat pouze omezenou sadu dat, co se cílového odvětví týče, tedy není nutné, aby byl přehnaně široký. Zároveň je potřeba brát v potaz možnost budoucího rozšíření, tedy využívaná technologie musí být schopna jednoduchého rozšíření.

Dalším požadavkem je jednoduchá správa aplikace. Užití technologie mají být jednoduché pro vývojáře, a to jak po stránce užívání, tak po stránce osvojení si dovedností v takovém softwaru.

S ohledem na skutečnost, že se jedná o studentský projekt, je žádoucí, aby užití technologie bylo možné použít jako freeware anebo případně pod univerzitní licenci. Za předpokladu, že by byl prototyp dále využíván v praxi, je open source preferován. Zde je ovšem potřeba dbát na to, zda tvůrce či poskytovatel licence zároveň nedistribuuje i placenou verzi svého produktu. V takovém případě je žádoucí užití takové verze systému, a to nejen z právního, ale i praktického hlediska.

Co se týče výběru mezi SQL a NoSQL řešení, je preferováno využití SQL řešení. I přes skutečnost, že NoSQL databáze, získávají na popularitě, a to hlavně z důvodu možnosti pracovat s libovolnou sadou dat, je potřeba připomenout, že klasické SQL databáze jsou stále zažitou praktikou a široká škála programátorů s nimi tedy umí pracovat.

## 4.2 Databázové Frameworky

V případě frameworků je na počátku vývoje nutné položit si otázku, zda je používat, či nikoliv. V první řadě záleží na komplexnosti systému. V případě jednodušších aplikací je možné vše programovat od počátku a nepoužívat framework. S přihlédnutím ke složitosti očekávané aplikace je jednodušší užít již existujících knihoven, které jsou již ve frameworkcích obsaženy, a které značně usnadní iniciaci vývoje samotného prototypu.

Zároveň je důležité brát v potaz cenu vývoje aplikace. V současnosti mnoho firem přistupuje k užívání frameworků, a to i například při vývoji her nebo aplikací pro telefony. Výhodou softwarového frameworku skutečnost, že jsou již předpřipravené a otestované. To mimo jiné zajišťuje, že užitý kód nebude obsahovat duplicitní složky, a že v něm bude minimum nadbytečných částí, což zvyšuje i spolehlivost samotného kódu. Takový kód je také považován za bezpečnější, a to i z pohledu konzistentnosti vývoje, který není do takové míry ohroženou chybou v kódu.

V neposlední řadě pak jejich užití výrazně zkracuje čas pro vývoj a také může snížit množství vydaných finančních prostředků.

Velké množství vývojářů do open source frameworků přispívá a všeobecně se do vývoje zapojuje, aby se navzájem podpořila v rámci celé komunity.

Využívání frameworků má ovšem i své nevýhody. Jako zásadní se jeví ta, že méně zkušení vývojáři pouze bezmyšlenkovitě užijí framework, jehož kódu nerozumí.

Porozumění kódu, na kterém systém funguje, je považováno za esenciální schopnost uživatele, a proto je těmto uživatelům doporučováno vyhnout se takovému řešení. V opačném případě nemusí být vývojář schopen reagovat na chyby či bugy, které mohou během vývoje nastat.

Důležité je také udržet konsistenci kódu, neboť jeho nekonzistence zvyšuje riziko prodlení a navyšuje cenu.

Softwarový framework, který bude užít k realizaci prototypu bude vybrán z následujících pěti.

### 4.2.1 Spring

Spring je open source framework, který je určen pro aplikační vývoj. Je zamýšlen pro platformu Java, a tedy je i v jazyce Java programován. Je určen pro vývoj aplikací na úrovni enterprise class.

Enterprise class je definováno jako aplikace, která je designována, aby byla robustní a škálovatelná na úrovni velké organizace. (Technopedia, 2021)

Předností tohoto frameworku je právě jeho jednoduchá škálovatelnost, a to jak při rozšiřování, tak při zmenšování.

Další předností Spring je takzvaná Dependency Injection. V zásadě jsou třídy v Java aplikacích vytvářeny tak, aby byly co nejvíce nezávislé na ostatních. To zvyšuje možnost opětovného užití. Díky tomu je možné je i lépe testovat. Zároveň je jistá míra provázanosti žádoucí, například při již zmíněném testování. K tomu slouží právě mechanismus Dependency injection, který různé třídy spojuje, a zároveň se je snaží držet co nevíce nezávislé. (Tutorialspoint, 2021)

Spring dále udržuje silnou zpětnou kompatibilitu. Spring podporuje pouze omezené množství knihoven třetích stran, aby zajistil maximální podporu knihoven a aplikací, které jsou na frameworku závislé. (Spring, 2021)

Spring také usnadňuje užití a integraci základních operací pod zkratkou CRUD:

- Create - Vložit
- Read - Číst
- Update - Aktualizovat
- Delete - Smazat

CRUD je takto integrován ve formě REST API, což je aplikační rozhraní, které figuruje jako prostředník mezi uživatelem a databází.

Spring framework není jediným produktem, který v rámci ekosystému Spring vznikl.

Mezi další Spring projekty patří:

- Boot
  - sada šablon, která umožňuje vytvářet projekty a aplikace založené na platformě Spring,
- Cloud
  - služba, která umožňuje vyvíjet postupy, které jsou specifické pro systémy rozložené mezi více serverů,
- Security

- spring poskytuje mechanismus, který umožňuje vyvinout ověřovací a autorizační procesy v rámci projektů, které jsou na platformě Spring založeny,
- Mobile
  - služba, která umožňuje rozpoznávání koncového zařízení, a následně upravuje výstup na základě daného zařízení,
- Batch
  - podporuje vývoj dávkových aplikací, které jsou určeny pro systémy větších společností. Takovou aplikací může být například archivace.

(Chandrakant, 2019)

Spring, co se jeho kvality týče, velmi spoléhá na komunitu, která je složena jak jednotlivých vývojářů, tak z jejich týmů, které na další vývoj dozírají. To může být považováno jak za výhodu, tak nevýhodu. Jako výhodu lze označit skutečnost, že komunita, která operuje na fórech, je schopná jednoduše pomoci jiným uživatelům.

Nevýhodou je samozřejmě to, že ne pro každý proces nebo knihovnu může existovat jednotný postup, případně manuál, jak v případě chyb nebo neznalosti uživatele postupovat.

Za nevýhodu lze považovat skutečnost, že Spring nepodporuje všechny specifikace Jakarta EE, tedy Java platform. Zároveň ovšem podporuje všechny jejich technologie. (Chandrakant, 2019)

#### 4.2.2 Micronaut

Jedná se o nejnovější framework, který vznikl na konci roku 2018, poté co vývojářský tým analyzoval konkurenční platformy jako Spring nebo Grail, na kterém členové skupiny také pracovali. Podobně jako Spring je vystavěn tak, aby dokázal spolupracovat s platformou Java. (Havers, 2019)

Micronaut je zaměřen a optimalizován na následující sadu aspektů:

- rychlá doba spuštění aplikace
- nízké nároky na paměť
- minimální využití reflexe a proxy
- minimum externích závislostí

- jednoduché a rychle testování aplikace

(Havers, 2019)

### 4.2.3 Hibernate

Hibernate je dalším frameworkem, který je programován v jazyce Java. Podporuje objektově relační mapování a ve spojení s JPA je doporučovaným řešením pro integraci standardů CRUD, podobně jako Spring. (Thorben Janssen, 2021)

V případě Hibernate byl pozorován problém datovým modelem many-to-many. Na základě toho je doporučováno raději užít dvě sady one-to-many, jeli možné je takto nahradit. Z pravidla v praxi existuje jeden prvek, který lze za tímto účelem využít.

Hibernate dokáže být pro nové uživatele těžké pro užívání. Abstrakt je velký a komplexní a uživatel musí trávit více času posuzováním konceptu, funkcí a užitím ve vývojovém programu. (Kihm, 2013)

## 4.3 Databázové systémy

Druhou potřebnou technologií jsou databázové systémy, ve kterých jsou data uchovávána a případně zpracovávána. Zde je nejprve potřeba rozlišit, zda užít SQL nebo NoSQL technologii. Obě technologie mají svá pozitiva i negativa.

Mezi pozitiva SQL databází, které lze označit také jako relační databáze, jednoznačně patří skutečnost, že jsou využívány již několik desítek let a programátoři takové technologie ovládají.

Další výhodou SQL databází je skutečnost, že dodržují takzvané 4 žádoucí vlastnosti ACID:

- Atomicita – transakce, které mohou být složeny z jednotlivých operací, jsou buď provedeny zcela nebo vůbec
- Konzistence – transakce, které v databázích probíhají nenarušují integritu samotné databáze
- Izolovanost – transakce se navzájem neovlivňují
- Trvalost – jakmile je transakce dokončena je trvale zaznamenána, a tedy její výsledek není vymazán například při výpadku sítě.

(Tobin, 2020)

Negativem takového systému je ovšem skutečnost, že není plně připraven na takzvaná Big Data, respektive není schopen plně pracovat s nestrukturalizovanými daty.



Další nevýhodou SQL databází je složitost jejich škálovatelnosti v případě růstu systému.

Naopak NoSQL databáze, která není závislé na striktní struktuře dat, je schopné zpracovávat různá data z rozdílných zdrojů o různé datové struktuře či kvalitě. Takto nestrukturalizovaná data lze použít například při strojovém učení, za účelem automatizace procesů, anebo mohou taková data skladovat a později je strukturalizovat. (Tobin, 2020)

V porovnání s SQL databázemi je škálovatelnost NoSQL databází jednodušší, a to hlavně proto, že jsou tyto databáze designované tak, aby bylo možné je snadno fragmentovat.

Při další komparaci s SQL jsou ovšem NoSQL systémy, relativně, novou technologií, a tedy nemusí s nimi být uživatelé plně seznámeni. Což může vést k nedostatečnému množství podpory.

Dalším problémem je nedostatek nástrojů, které pomáhají s testováním výkonu a analýzou. (Tobin, 2020)

Poslední uvedenou nevýhodou NoSQL databází je složitost standardizace a kompatibility. Obojí vyplývá ze skutečnosti, že dané databáze pracují s nestrukturalizovanými údaji, což může komplikovat propojování databází s jinými službami nebo produkty dané společnosti.

Existují tři faktory, které je nutné zohlednit v případě výběru databázového systému. Jmenovitě uvádí Tobin tři, a to:

- Atomicita
  - Pokud je prioritou uživatele atomicita, tedy stav, kdy proběhne transakce úplně nebo vůbec, má uživatel volit relační databázi, a to na základě skutečnosti, že dodržuje vlastnosti ACID.
- Vertikální a horizontální škálovatelnost
  - Škálovatelnost je dle (Holubová, 2015, s.48) z hlediska problematiky zpracování dat vlastnost systému (popř. sítě, procesu, apod.) flexibilně reagovat na měnící se požadavky, v našem případě zejména na zvyšující se objemy dat a zátěž systému.
  - V případě, kdy je očekávána pouze vertikální škálovatelnost, tedy databáze je rozšiřována pouze navyšováním výpočetní kapacity serveru, je relační databáze dostačující technologií. Mezi výhody vertikální škálovatelnosti patří jednoduchost a rychlost.

- Naopak v případě horizontálního škálování, k tomu dochází při nárůstu uživatelů nebo dotazů, a to formou navyšování počtu serverů, je doporučeno užití nerelačních databází. V principu se tedy uživatelé nebo požadavky rovnoměrně rozprostřou do více serverů.
- Rychlost
  - V situaci, kdy je pro uživatele přednější rychlost transakcí a jejich zpracování, před dodržením ACID, je doporučována nerelační databáze. Toto řešení se uplatňuje zejména v operacích real-time daty, kdy nedokonalá integrita dat může být tolerována ve prospěch rychlosti.

(Tobin, 2020)

Mezi porovnávané databázové systémy jsou zařazeny jak SQL, tak i NoSQL databáze. Jmenovitě se jedná o MySQL, Oracle, Microsoft SQL Server a PostgreSQL, které jsou zástupci SQL databází, a IBM DB2 a MongoDB, které jsou zástupci NoSQL databází.

#### 4.3.1 MySQL

Jde o relační databázový systém, který je psát v programovacích jazycích C a C++. Je nabízen pod dvěma licencemi. Těmi jsou open source MySQL Community Server, který je podporován právě komunitou, a proprietární Enterprise Server, u kterého již uživatel hradí poplatky za předplatné, licenci, podporu a případné zaškolení. (DBQuest, 2021)

Provozovatelem MySQL je společnost Oracle. Ten svým uživatelům zajišťuje nepřetržitou podporu ve všech fázích nasazení i provozování databázového systému.

Podobně jako další databázové systémy je MySQL jednoduše škálovatelný a flexibilní do té míry, že si jej, v open source verzi, může uživatel upravovat, jak potřebuje. Dále pak podporuje ACID.

Další výhodou systému je unikátní architektura úložného modulu, která umožňuje databázovým profesionálům nakonfigurovat databázový server MySQL specificky pro dané aplikace. (DBQuest, 2021)

MySQL disponuje vysoce výkonným modulem pro dotazování, díky kterému je označován za standart pro webové stránky s vysokou návštěvností.

Dále MySQL nabízí bezpečnostní funkce, které zajišťují ochranu dat. Mezi ty patří mechanismus, který zajišťuje, že přístup k serveru mají pouze autorizované osoby.

MySQL je často užíván jako jedna z komponent systémového balíčku LAMP, který je určen pro vývoj dynamických webových stránek a webových aplikací. LAMP je akronym, který je složen z prvních písmen názvů užitých systémů. Jmenovitě Linux, v roli operačního systému, Apache jako webový software, MySQL jako databázový systém a, nejčastěji, PHP jako programovací jazyk. (Atlantic.Net, 2021)

### 4.3.2 Oracle Database

Dalším relačním databázovým systémem je Oracle Database, vyvíjený a spravovaný společností Oracle. Je napsán v programovacích jazycích C a C++. Podobně jako v případě jiných databázových systémů je do Oracle implementován objektově relační model za účelem skladování komplexních business modelů v relační databázi. (Oracle Help Center, 2021)

Za účelem udržení kvality dat je Oracle Database v souladu s ACID. Databáze je zároveň cross-platform, tedy dokáže fungovat na varietě operačních systémů včetně Windows, Unix nebo Linux. (Oracle Tutorial, 2021)

Oracle Database je relační databázový systém s největším podílem na trhu. (Oracle Tutorial, 2021)

Jedním z nástrojů, kterým Oracle Database disponuje, je Recovery Manager, který umožňuje z databáze vytvářet zálohy a takzvanou point-in-time obnovu. Ta umožňuje vrátit databázi do stavu, ve kterém byla ve vybraném bodě v čase.

Společnost Oracle nabízí tři varianty Oracle Database, a to:

- Enterprise Edition
  - Na úrovni poskytnuté výpočetní kapacity, operační paměti nebo velikosti databáze není omezena
- Standard Edition
  - Výpočetní kapacita je omezena na maximálně čtyři procesory, operační paměť ani velikost databáze není omezena
- Expression Edition
  - Výpočetní kapacita je omezena na dva procesory, operační paměť RAM je omezena na 2 GB, velikost databáze je pak omezena na 12 GB uživatelských dat

(Oracle Tutorial, 2021)

### 4.3.3 Microsoft SQL Server

Relační databázový systém vyvíjený a spravovaný společností Microsoft, který je napsán v programovacích jazycích C a C++. Jako předchozí databáze implementuje objektivě relační model.

Microsoft SQL serveru je určen pro užití při analýze dat, generování reportů, a integraci dat za pomoci operací ETL. (Tutorialspoint, 2021)

Výhodami Microsoft SQL serveru jsou:

- Podpora mobility a mobilních zařízení
  - Zprávu o vývoji a vizualizaci dat lze získat za pomoci mobilních zařízení
- Integrace s dalšími produkty Microsoft
  - To je výhodou pro společnosti, které silně spoléhají na další produkty od společnosti Microsoft
- Rychlost
  - Rychlost a stabilita jsou dvě vlastnosti, na kterých Microsoft SQL Server vybudoval svoji reputaci

(Tobin, 2020)

Jako nevýhodu Tobin vidí cenu, která přesahuje 14 000 dolarů za jádro, v případě licence pro velké společnosti. Microsoft ovšem nabízí i dvě levnější varianty a bezplatnou verzi, díky které si uživatel může vyzkoušet práci s tímto systémem. Další zmiňovanou nevýhodou jsou velké požadavky na zdroje ve smyslu hardware. (Tobin, 2020)

### 4.3.4 PostgreSQL

PostgreSQL je open source objektivě-relační databáze, která podporuje dotazování za pomoci SQL i JSON, tedy jak relační, tak nerelační. Zároveň se řadí mezi takzvané Enterprise Class databáze, tedy je možné ji využít pro velké společnosti. Dále je v souladu s ACID, podobně jako jiné relační databáze. Je vyvinut tak, aby fungoval na všech hlavních operačních systémech. (PostgreSQL Tutorial, 2021)

I přes skutečnost, že PostgreSQL je napsán v programovacím jazyce C, podporuje většinu momentálně používaných jazyků, jako například, Java, JavaScript, Python, C# a mnoho dalších.

PostgreSQL je vysoce rozšiřitelný. Jako příklad uvádí možnost definovat vlastní datové typy, vystavovat vlastní funkcionality, a možnost psát kód v různých programovacích jazycích bez nutnosti rekompile databáze. (PostgreSQL, 2021)

Tvůrci systému se snaží, aby PostgreSQL splňoval všechny SQL standarty, ovšem v některých případech dochází k lehké změně syntaxe nebo funkce. Verze 13, která byla nasazena v září roku 2020 vyhovuje asi 170 ze 179 povinných vlastností dle SQL:2016 Core. Dále pak tvrdí, že v současné době žádná relační databáze nesplňuje všechny standarty. (PostgreSQL, 2021)

Výhodou PostgreSQL je jeho vysoká škálovatelnost. Tu lze pozorovat jak v případě množství dat, se kterým je databáze schopna operovat, tak v počtu různých uživatelé, které je schopna obsluhovat. (Tobin, 2020)

Další výhodou databáze je její univerzálnost, tedy ji lze použít za, téměř, jakýmkoli účelem. To lze zároveň považovat za nevýhodu, neboť není v základní verzi nijak specializovaná. Ovšem díky široké škále možností následných úprav se nemusí jednat problém.

#### **4.3.5 MongoDB**

Jedná se o jediný čistě NoSQL nerelační databázový systém, který je v seznamu uveden. Dokumentový databázový model, který MongoDB užívá, je silnou cestou, jak ukládat a načítat data, což vývojářům pomáhá pracovat rychleji. (MongoDB, 2021)

Mongo nabízí jak komunitní, bezplatnou, edici svého systému, která je distribuovaná jako open-source, tak firemní edici, kterou lze využívat například v datových centrech. (MongoDB, 2021)

Světově je MongoDB podporován širokou komunitou, takže není těžké získat pomoc, v případě problémů. Zároveň MongoDB podporuje všechny hlavní programovací jazyky, což usnadňuje jeho užívání.

Za největší nevýhodu MongoDB, potažmo všech NoSQL databází, lze pokládat, že k nim neexistuje velké množství definovaných standardů, což vede k problému, kdy dvě NoSQL databáze mohou být naprosto odlišné. To samozřejmě může nastat i u SQL databází. Možnost, že taková situace nastane, je ovšem značně omezena díky době, po kterou jsou SQL databáze využívány, a za kterou bylo možné vytvořit k takovým databázím všeobecné standarty. (Sai Krishna, 2019)

MongoDB je jednou z nejvíce užívaných open source NoSQL databází, založených na dokumentovém modelu. (Sharma, 2020)

Z podstaty NoSQL databází, kdy takové databáze umí pracovat s velkým množstvím nestrukturalizovaných dat, je správné je užívat pro práci s big data a v odvětvích, kdy je rychlost zpracování dat preferována před kvalitou. NoSQL databáze mají tak často potíže splňovat vlastnosti ACID.

Dle Tobina je nevýhodou MongoDB složitost nastavení. Je na něj potřeba více času a zkušeností než v případě užití jiných řešení. Další nevýhodou, o které Tobin píše, jsou problémy s bezpečností. Nativní nastavení MongoDB totiž není natolik bezpečné, a proto je potřeba podniknout návazné kroky, které bezpečnost databáze zajistí. (Tobin, 2020)

#### 4.4 Vícekriterální analýza variant

Pro výběr ideální varianty, pro tvorbu prototypu, byla zvolena vícekriteriální analýza, respektive metoda TOPSIS.

V případě výběru databázového framework jsou zkoumány následující parametry:

- Rozšířenost
  - množství uživatelů, kteří daný framework využívají,
- Stáří
  - jsou preferovány spíše starší databázové frameworky, u kterých je předpoklad celkového odladění a vspělosti softwaru.

Tabulka 1 Kriteriační matice frameworků

	Rozšíření	Stáří
Spring	782	2002
Hibernate	176	2001
Micronaut	9	2018
Váhy	0,6	0,4
Z	801,6115019	3476,252149
Povaha	MAX	MIN

Zdroj: Vlastní zpracování, 2021

Tabulka 2 Normalizovaná matice frameworků

	Rozšíření	Stáří
Spring	0,585320943	0,230363036
Hibernate	0,131734637	0,23024797
Micronaut	0,00673643	0,2322041
Povaha	MAX	MIN
h (ideální)	0,585320943	0,23024797
d (bazální)	0,00673643	0,2322041

Zdroj: Vlastní zpracování, 2021

Data pro výčet ideální varianty frameworku byla čerpána, z výše zmíněných zdrojů a informací, z oficiálních stránek poskytovatelů, případně webu stackshare.io.

Pro výběr databázových systémů jsou nastaveny následující parametry:

- Cena
  - preferována je nejnižší cena,
- Škálovatelnost
  - preferována je možnost škálovatelnosti, a to jak horizontální, tak vertikální,
- Dokumentace
  - parametrem dokumentace se rozumí celková dostupnost, podpora a srozumitelnost poskytnuté dokumentace,
- Learning curve
  - míra zdatnosti uživatele v poměru se zkušeností.

Tabulka 3 Kriteriační matice databází

	Cena	Škálovatelnost	Dokumentace	Learning Curve
MySQL	5000	3	5	7
Oracle	47000	5	7	3
PostgreSQL	0	5	5	5
MSSQL Server	14000	3	7	3
MongoDB	20000	5	7	7
Váhy	0,5	0,2	0,2	0,1
Z	53197,74431	9,643650761	14,03566885	11,87434209
Povaha	MIN	MAX	MAX	MAX

Zdroj: Vlastní zpracování, 2021

Tabulka 4 Normalizovaná matice databází

	Cena	Škálovatelnost	Dokumentace	Learning Curve
MySQL	0,046994474	0,062217102	0,07124705	0,058950634
Oracle	0,441748053	0,103695169	0,09974587	0,025264558
PostgreSQL	0	0,103695169	0,07124705	0,042107596
MSSQL Server	0,131584527	0,062217102	0,09974587	0,025264558
MongoDB	0,187977895	0,103695169	0,09974587	0,058950634
Povaha	MIN	MAX	MAX	MAX
h (ideální)	0	0,103695169	0,09974587	0,058950634
d (bazální)	0,441748053	0,062217102	0,07124705	0,025264558

Zdroj: Vlastní zpracování, 2021

Data pro výčet ideální varianty databází byla čerpána z výše zmíněných zdrojů a informací, oficiálních stránek poskytovatelů, případně webu altexsfot.com.



## 5 Výsledky a diskuse

Tabulka 5 Výpočet pořadí frameworků

	d+	d-	c	Pořadí
Spring	0,000115066	0,578587442	0,999801165	1
Hibernate	0,453586306	0,125013511	0,216062134	2
Micronaut	0,578587819	0	0	3

Zdroj: Vlastní zpracování, 2021

Tabulka 6 Výpočet pořadí databází

	d+	d-	c	Pořadí
MySQL	0,068855598	0,396188264	0,851937411	2
Oracle	0,44303058	0,050325072	0,102005666	5
PostgreSQL	0,033103938	0,444010654	0,93061638	1
MSSQL Server	0,142019962	0,311470057	0,686828914	3
MongoDB	0,187977895	0,260895876	0,581223258	4

Zdroj: Vlastní zpracování, 2021

Na základě získaných výsledků analýz metodou TOPSIS bylo vyhodnoceno, že jako řešení, které se nejvíce blíží vytyčeným požadavkům, je využití kombinace Spring jako frameworku a PostgreSQL jako databázového systému. PostgreSQL je dostatečně uživatelsky přívětivý, zatímco Spring umožňuje snadno upravovat různé funkcionality.

Jako programovací jazyk byla zvolena Java, jako jazyk, který je pro obě technologie určen.

Pro výstavbu databáze byla užito takzvané Layered architecture. Tento typ architektury umožňuje, díky tomu, že jsou jednotlivé vrstvy na sobě nezávislé, jednodušší testování a zároveň je jednoduchá na implementaci.

Skládá se ze čtyř hlavních vrstev:

- databázové, ve které se uchovávají data,
- persistentní, která spravuje funkce jako ORM,
- business, obsahuje business logiku,
- prezentační, ta obsahuje kategorie s touto vrstvou spjaté.

Nevýhodou ovšem může být skutečnost, že při změnách je potřeba ji nasazovat jako celek, což může dočasně omezovat provoz databáze.

Pro komunikace mezi uživatelem a systémem je užito rozhraní REST API. Umožňuje přistupovat k datům pomocí standardních metod http, a zároveň jsou pomocí něj implementovány metody CRUD.

Za účelem aplikace objektově relačního mapování je dále užita Java Persistent API (JPA).

Jako autentizační metoda je použit JSON Web Token (JWT), který zajišťuje, že se do databáze nemůže připojit uživatel, který k takové akci není oprávněn.

Prototyp aplikace disponuje následujícími funkcemi ve třech hlavních skupinách:

- Uživatel aplikace
  - autentikace uživatele,
  - připojení uživatele,
- Klient uživatele
  - vytvoření klienta,
  - aktualizace klienta,
  - výmaz klienta,
  - vyvolání všech klientů,
  - vyvolání klienta podle jeho unikátního id,
  - vyvolání kontraktů klienta podle jeho unikátního id,
- Kontrakt klienta
  - vytvoření kontraktu,
  - aktualizace kontraktu.

Pro spuštění prototypu aplikace je nutné, aby uživatel na svém počítači měl nainstalovanou Javu, alespoň ve verzi 11, aplikaci Maven, a databázový systém PostgreSQL. Dále pak ještě samotnou aplikaci, která je k bakalářské práci přiložena ve formě složky ve formátu .zip pod názvem BCP-main.

## 5.1 Diskuse

Kvůli skutečnosti, že je databáze vytvářena jako SQL, je nutné umístit striktnější pravidla s ohledem na kvalitu a strukturu umístitovaných dat. To může být problém v případě importu dat z domácích databází uživatelů, které se mohou strukturně lišit.

V budoucnu je možné očekávat větší užití NoSQL technologie, která není striktní do takové míry a již nyní získává na popularitě. Spolu s tím je také NoSQL databáze lepším řešením pro Big data, a tedy v momentě, kdy by byl předpoklad užití databáze pro takto velké množství dat, bylo by odpovídajícím použít právě tuto technologii.

Dalším důležitým aspektem, je čistota, respektive čitelnost kódu. Ta se zvyšuje se zkušeností tvůrce a množstvím aplikací, případně kódu, na kterých tvůrce pracoval.

## 6 Závěr

S ohledem ke stávající finanční situaci občanů, která je ovlivněna pandemií onemocnění COVID-19, je možné počítat s tím, že bude vznikat větší množství podobných registrů. Společně s tím je možné, že jednotliví poskytovatelé mikroúvěrů, nebo i jiných odvětví, dospějí k názoru, že mohou daný registr vytvořit z vlastní iniciativy, a posléze v něm sdružovat více subjektů a jejich dat. V takovém případě by se ovšem nemuselo jednat o komerční registr, a uživatelé by nemuseli hradit poplatky za dotazování, ale pouze za jeho údržbu, respektive provoz.

Nevýhodou takového kroku je ovšem nutnost najmu specialistu, který požadovaný systém navrhne. Na základě jeho návrhu je pak možné systém vybudovat. Na výstavbu systému je nutné si opět najmout skupinu vývojářů, kteří rozumí dané problematice a zároveň ovládají vybrané systémy. Alternativou je vyčlenit tým z vlastních zdrojů, který se bude vývoji věnovat.

V prvním případě může společnost, která systém požaduje, čelit zvýšeným nákladům za najatého specialistu a tým vývojářů.

V druhém případě může nastat situace, při které vyčleněný tým bude chybět v klasickém provozu, což může vést ke snížení kvality stávajících služeb, po dobu vývoje nového systému.

Finanční registry jsou a pravděpodobně i budou stále relevantní, a to nejen pro poskytovatele finančních služeb. V současné době lze pozorovat rozšiřování členskýchází registru i o poskytovatele energií, případně telekomunikací, kteří data mohou využívat v případě vymáhání pohledávek. Příkladem může být Registr platebních informací, který dle jejich webových stránek operuje s daty sedmi energetických společností. (REPI, 2021)

Informační tok ovšem může fungovat i obráceně, a to tím způsobem, že poskytovatelé finančních služeb se mohou lépe rozhodovat o poskytnutí či neposkytnutí služby, na základě širší škály dat z více odvětví.

Pro uživatele registrů je výhodou množství nákladů, které se, bez nutnosti registr vystavět a provozovat, skládají pouze z poplatků za dotazování a případných poplatků za zpřístupnění.

Atraktivitu registrů jako informačních zdrojů lze pozorovat na překryvu jejich uživatelů. Uživatelé totiž často využívají více registrů, aby zkvalitnili svůj rozhodovací proces. Příkladem takto komplementárních registrů jsou REPI a NRKI.

## 7 Seznam použitých zdrojů

### *Knížní publikace*

- BROŽOVÁ, Helena, Milan HOUŠKA a Tomáš ŠUBRT. *Modely pro vícekriteriální rozhodování*. Praha: Credit, 2003. ISBN 978-80-213-1019-3.
- HOLUBOVÁ, Irena, Jiří KOSEK, Karel MINAŘÍK a David NOVÁK. *Big Data a NoSQL databáze*. Praha: Grada, 2015. Profesionál. ISBN 978-80-247-5466-6.
- KNIGHT, Brian. *Microsoft SQL Server 2000: pokročilé techniky*. Brno: Computer Press, 2004. ISBN 80-251-0111-8.
- KUČERA, Petr. Efektivní posuzování platební morálky závisí na schopnosti vytěžování nových typů dat. *Bankovníctví*. Praha: *Economia*, 2020, **27**(1/2020), 34-35. ISSN 1212-4273.
- MEJSTŘÍK, Michal, Magda PEČENÁ a Petr TEPLÝ. *Bankovníctví v teorii a praxi: Banking in theory and practice*. Praha: Karolinum, 2014. ISBN 978-80-246-2870-7.
- MOMJIAN, Bruce. *PostgreSQL: praktický průvodce*. Brno: Computer Press, 2003. ISBN 80-7226-954-2.
- ONDŘEJ, Jan. *Spotřebitelské smlouvy a ochrana spotřebitele: ekonomické, právní a sociální aspekty*. Praha: C.H. Beck, c2013. Beckova edice právní instituty. ISBN 978-80-7400-446-9.
- PROCHÁZKA, David. *Oracle: průvodce správou, využitím a programováním nad databázovým systémem*. Praha: Grada, 2009. Průvodce (Grada). ISBN 978-80-247-2762-2.
- PROKOPOVÁ, Zdenka. *Databázové systémy MySQL+PHP*. Ve Zlíně: Univerzita Tomáše Bati, 2006. ISBN 80-7318-486-9.
- ŠTĚDRONĚ, Bohumír. *Teorie a praxe strategického a manažerského řízení v ICT*. Davle: Kernberg, 2009. Basic. ISBN 978-80-87168-13-4.
- TYRYCHTR, J. *Provozní a analytické databáze - Teoretické základy*. Praha : ČSVIZ, 2015. ISBN 978-80-87968-02-4.
- *Sbírka zákonů Česká republika*. Břeclav: Moraviapress. ISSN 1211-1244.

### *Webové stránky a příspěvky na webových stránkách*

- *Atlantic.Net* [online]. [cit. 2021-02-22]. Dostupné z: <https://www.atlantic.net/>

- BOTELHO, Britget a Jack VAUGHAN. MongoDB. *TechTarget* [online]. 2020 [cit. 2021-02-22]. Dostupné z: <https://searchdatamanagement.techtarget.com/definition/MongoDB>
- *CBCB.cz - Bankovní registr klientských informací* [online]. [cit. 2021-03-10]. Dostupné z: <https://www.cbcb.cz/>
- Centrální registr úvěrů. *Česká národní banka* [online]. [cit. 2021-03-10]. Dostupné z: <https://www.cnb.cz/cs/dohled-financni-trh/centralni-registr-uveru/>
- *CNCB - Nebankovní registr klientských informací* [online]. [cit. 2021-03-10]. Dostupné z: <https://www.cncb.cz/>
- *CRIF - Czech Credit Bureau* [online]. [cit. 2021-03-10]. Dostupné z: <https://www.crif.cz/>
- *DBQuest, Inc.* [online]. [cit. 2021-03-01]. Dostupné z: <http://www.databasequest.com/>
- HAVERS, Jonas. Building Microservices with Micronaut. *Oracle* [online]. 2019 [cit. 2021-03-01]. Dostupné z: <https://blogs.oracle.com/javamagazine/building-microservices-with-micronaut>
- *Hibernate* [online]. [cit. 2021-03-10]. Dostupné z: <https://hibernate.org/>
- HUGHES, Adam a Craig STEDMAN. Microsoft SQL Server. *TechTarget* [online]. 2019 [cit. 2021-02-26]. Dostupné z: <https://searchsqlserver.techtarget.com/definition/SQL-Server>
- CHANDRAKANT, Kumar. Why Choose Spring as Your Java Framework? *Baeldung* [online]. 2019 [cit. 2021-03-01]. Dostupné z: <https://www.baeldung.com/spring-why-to-choose>
- Comparing Database Management Systems: MySQL, PostgreSQL, MSSQL Server, MongoDB, Elasticsearch and others. *AltexSoft* [online]. [cit. 2021-03-10]. Dostupné z: <https://www.altexsoft.com/blog/business/comparing-database-management-systems-mysql-postgresql-mssql-server-mongodb-elasticsearch-and-others/>
- KIHM, Frederic. Using Hibernate Frameworks: What Are the Best Practices? *CAST* [online]. 2013 [cit. 2021-03-01]. Dostupné z: <https://www.castsoftware.com/blog/using-hibernate-frameworks-what-are-the-best-practices>

- *Micronaut Framework* [online]. [cit. 2021-03-10]. Dostupné z: <https://micronaut.io/index.html>
- *Microsoft* [online]. [cit. 2021-03-10]. Dostupné z: <https://www.microsoft.com/cs-cz/>
- *MongoDB* [online]. [cit. 2021-03-01]. Dostupné z: <https://www.mongodb.com/>
- *MySQL* [online]. [cit. 2021-03-10]. Dostupné z: <https://www.mysql.com/>
- *Oracle* [online]. [cit. 2021-03-10]. Dostupné z: <https://www.oracle.com/index.html>
- *Oracle Help Center* [online]. [cit. 2021-03-01]. Dostupné z: <https://docs.oracle.com/en/>
- *Oracle Tutorial* [online]. [cit. 2021-03-01]. Dostupné z: <https://www.oracletutorial.com/>
- *PostgreSQL* [online]. [cit. 2021-03-01]. Dostupné z: <https://www.postgresql.org/>
- *PostgreSQL Tutorial* [online]. [cit. 2021-03-01]. Dostupné z: <https://www.postgresqltutorial.com/>
- REDDEM, Sai Krishna. When to Use MongoDB Rather than MySQL. *Medium* [online]. 2019 [cit. 2021-03-01]. Dostupné z: <https://medium.com/@rsk.saikrishna/when-to-use-mongodb-rather-than-mysql-d03ceff2e922>
- *REPI - Registr platebních informací* [online]. [cit. 2021-03-10]. Dostupné z: <https://www.repi.cz/>
- *SOLUS - Zájmové sdružení právnických osob* [online]. [cit. 2021-03-10]. Dostupné z: <https://www.solus.cz/>
- SHARMA, Kris. NoSQL databases: what is MongoDB and its use cases? *Ubuntu* [online]. 2020 [cit. 2021-03-10]. Dostupné z: <https://ubuntu.com/blog/nosql-databases-what-is-mongodb-and-its-use-cases>
- *Spring* [online]. [cit. 2021-03-01]. Dostupné z: <https://spring.io/>
- *StackShare* [online]. [cit. 2021-03-10]. Dostupné z: <https://stackshare.io>
- *Technopedia* [online]. [cit. 2021-03-01]. Dostupné z: <https://www.techopedia.com/>
- *TheServerSide* [online]. [cit. 2021-02-15]. Dostupné z: <https://www.theserverside.com/>
- *Thorben Janssen* [online]. [cit. 2021-03-01]. Dostupné z: <https://thorben-janssen.com/>

- *ThoughtWorks* [online]. [cit. 2021-02-15]. Dostupné z:  
<https://www.thoughtworks.com/>
- TOBIN, Donald. Which Modern Database Is Right for Your Use Case? *Xplenty* [online]. 2020 [cit. 2021-03-01]. Dostupné z:  
<https://www.xplenty.com/blog/which-database/>
- *Tutorialspoint* [online]. [cit. 2021-03-01]. Dostupné z:  
<https://www.tutorialspoint.com/index.htm>
- What is MongoDB? *Guru99* [online]. [cit. 2021-02-26]. Dostupné z:  
<https://www.guru99.com/what-is-mongodb.html>