

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačních technologií**



**Bakalářská práce**

**Návrh relační databáze**

**Oldřich Barták**

© 2022 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Oldřich Barták

Informatika

Název práce

**Návrh relační databáze**

Název anglicky

**Relational database design**

---

## Cíle práce

V práci bude popsáno jak navrhnout a zároveň optimalizovat relační databázi v Oracle světě, využívanou firmou pro klientská data a jejich analytické zpracování. Data dále budou sloužit jako podklad k informovanému rozhodnutí.

## Metodika

K optimalizaci databáze bude využita Normalizace databáze, díky které budeme schopni i databázi dále udržovat při jejím rozšiřování o další data. Databázi je nutné stále rozšiřovat i kvůli novým požadavkům a výsledkům práce DataScience. Dále bude partitioning k rychlejšímu přístupu ke specifickým částím tabulek, bude využit zejména u tabulek s větším objemem dat a frekventovanou změnou.

**Doporučený rozsah práce**

30-40

**Klíčová slova**

Oracle, Relační databáze, partitioning, SQL

---

**Doporučené zdroje informací**

LACKO, Ľ. *Oracle : správa, programování a použití databázového systému*. Brno: Computer Press, 2002. ISBN 80-7226-699-3.

VOSTROVSKÝ, V. – ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE. KATEDRA INFORMAČNÍHO INŽENÝRSTVÍ. *Vytváření databází v ORACLE*. V Praze: Česká zemědělská univerzita, Provozně ekonomická fakulta, 2004. ISBN 80-213-1191-6.

---

**Předběžný termín obhajoby**

2022/23 LS – PEF

**Vedoucí práce**

Ing. Martin Pelikán, Ph.D.

**Garantující pracoviště**

Katedra informačního inženýrství

---

Elektronicky schváleno dne 27. 2. 2023

**Ing. Martin Pelikán, Ph.D.**

Vedoucí katedry

---

Elektronicky schváleno dne 2. 3. 2023

**doc. Ing. Tomáš Šubrt, Ph.D.**

Děkan

V Praze dne 05. 03. 2023

### **Čestné prohlášení**

Prohlašuji, že svou bakalářskou práci "Návrh relační databáze" jsem vypracoval(a) 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(ka) uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 15.3.2023



## **Poděkování**

Rád(a) bych touto cestou poděkoval(a) Ing. Martin Pelikán, Ph.D. za vedení práce a pomoc se zaměřením práce. Nadě Pernerové za korekci a kontrolu práce.

# Návrh relační databáze

## Abstrakt

Práce pojednává o práci, tvorbě a optimalizaci relačních databází. Je zde snaha popsat problémy vznikající při práci s relační databází, která se stále rozrůstá s nově vznikajícími daty a potřebami uživatelů využívajících reportů postavených právě nad relačními databázemi. Relační databáze jsou stále základními kameny, na těchto kamenech se následně staví nový svět reportů a velkých dat.

Samotná struktura jednotlivých tabulek v relačních databázích má velký dopad na efektivitu jednotlivých reportů postavených na těchto tabulkách. Proto si popíšeme, jak je dobré uvažovat pro navrhování jednotlivých tabulek a jak jednotlivé věci mohou ovlivňovat věci navázané na vaše tabulky.

**Klíčová slova:** Oracle, relační, databáze, partitioning, SQL, reporty, optimalizace, hints

# Relational database design

## Abstract

The thesis deals with the work, creation and optimization of relational databases. There is an attempt to describe the problems arising when working with relational databases, which are constantly growing with new emerging data and the needs of users using reports built on top of relational databases. Relational databases are still the cornerstones, and it is on these cornerstones that the new world of reports and big data is subsequently built.

The very structure of individual tables in relational databases has a big impact on the effectiveness of individual reports built on top of these tables. Therefore, we'll describe how to think about designing individual tables and how individual things can affect things tied to your tables.

**Keywords:** Oracle, relational, database, partitioning, SQL, reports, optimization, hints

# Obsah

<b>1. Úvod</b> .....	<b>10</b>
<b>2. Teoretická východiska</b> .....	<b>12</b>
1.1 Představení relačních databází .....	12
1.1.1 Normalizace relační databáze .....	13
1.1.1.1 První normalizační forma .....	14
1.1.1.2 Druhá normalizační forma .....	14
1.1.1.3 Třetí normalizační forma .....	14
1.1.2 Možnosti optimalizace .....	15
1.2 ERD model.....	17
1.3 Reportování dat .....	19
1.3.1 Co je data reporting?.....	19
1.3.2 Jak vytvářet datové reporty .....	20
1.3.3 Jak správně zobrazit data .....	21
<b>3. Vlastní práce</b> .....	<b>23</b>
1.4 Popsání datového modelu .....	23
1.4.1 Tabulka Klienti .....	24
1.4.2 Tabulka ESS .....	26
1.4.3 Tabulka Produkty.....	26
1.4.4 Tabulka Produkty_mesic .....	29
1.4.5 Tabulka Pobočka.....	31
1.4.6 Tabulka Klienti_fakta .....	32
1.4.7 Tabulka Adresy .....	33
1.4.8 Tabulka Obsluha .....	34
1.4.9 Tabulka Zdroj_sys .....	35
1.4.10 Tabulka Segmentace_klienta .....	36
1.4.11 Tabulka Segmetnace_fakta .....	38
1.4.12 Tabulka Obor_podnikani .....	38
<b>Výsledky a diskuse</b> .....	<b>38</b>
1.5 Původní návrh databáze .....	39
1.5.1 Rozdílné pohledy .....	41
<b>Závěr</b> .....	<b>42</b>
<b>4. Seznam použitých zdrojů</b> .....	<b>43</b>
<b>5. Bibliografie</b> .....	<b>43</b>
<b>6. Seznam obrázků, tabulek, grafů a zkratk</b> .....	<b>44</b>



1.6	Seznam obrázků .....	44
1.7	Seznam tabulek .....	44

# 1. Úvod

Aktuální trendem je data driven prostředí, na to je potřeba mít dostupná data nad kterými pak lze dělat rozhodnutí. Abychom mohli získávat a zpracovávat kvalitní data, potřebujeme mít navrhnuté a vyvinuté tabulky v databázi. Toto jsou tedy témata, kterým se v této práci budu věnovat a popisovat, jak i stávající řešení připravit a optimalizovat pro další práci s daty. Důležitým předpokladem je práce v prostředí, kde k datům přistupuje více uživatelů zároveň a nejedná se tedy o lokální úložiště.

Ve většině případů se totiž nestaví řešení pouze na „zelené louce“, ale spíš již využíváme nějakého stávajícího řešení. Zároveň musíme počítat s tím, že zdrojem dat jsou také již nějaké zavedené systémy, které mají svá omezení a způsoby předávání dat. V neposlední řadě musíme vzít v potaz jak a v čem budeme data dále zpracovávat a zobrazovat reporty, protože to nám přináší další výzvy v přístupu k datům

Pro naše potřeby budeme tvořit databázi firemních klientů pro nejmenovanou společnost působící ve finančním sektoru jejíž součástí je i sektor firemních klientů, se kterými zde budeme pracovat.

## Cíl práce a metodika

Cílem této práce je návrh přístupu k situaci, kdy z malého datového úložiště určenému k optimalizaci databáze bude využita Normalizace databáze, díky které ji budeme schopni dále udržovat při jejím rozšiřování o další data. Databázi je nutné stále rozšiřovat i kvůli novým požadavkům a výsledkům práce DataScience. Následuje partitioning, který slouží k rychlejšímu přístupu ke specifickým částem tabulek. Využívat ho budeme zejména u tabulek s větším objemem dat a frekventovanou změnou.

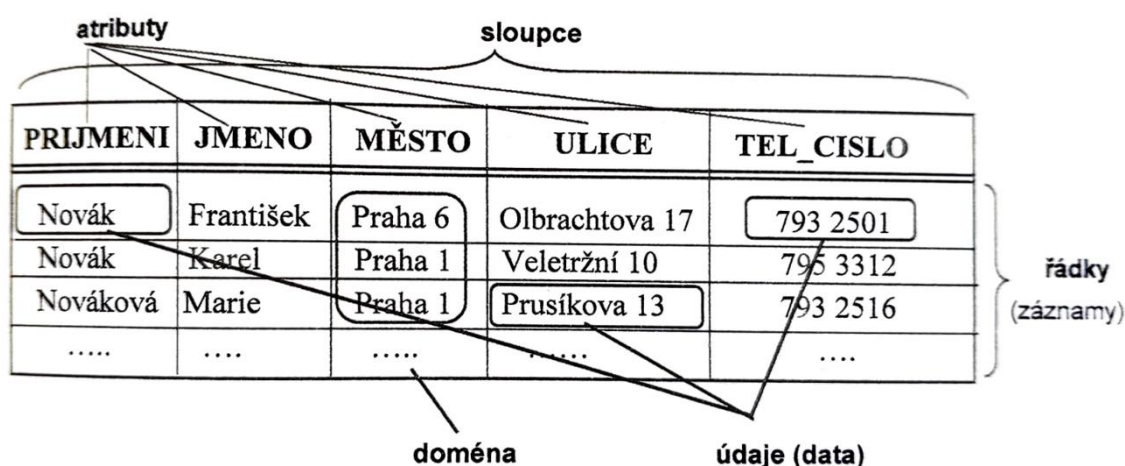
## 2. Teoretická východiska

Teoretická část popisuje relační databáze a jejich funkci. Tato část je nutná z důvodu správného pochopení možností a omezení, které relační databáze obsahují.

Relační databáze mají totiž vždy daná pravidla, která se musí z podstaty věci dodržovat. Čím lépe si tedy připravíme strukturu, nebo naopak ji zpětně upravíme dle požadavků, které nám vyvstanou při analýze aktuálního stavu, tím lépe.

### 1.1 Představení relačních databází

Relační databáze jsou základním stavebním kamenem všech aplikací, které pracují s daty. Je jedno jestli aplikace využívají data pouze pro zobrazení nebo pro čtení ve většině případů budou využívat řešení relačních databází. Každá aplikace je totiž nějak strukturována a vyžaduje jasně daná pravidla, a to nám pomáhá i při tvorbě relačních databází. Relační databáze také vychází z jasně daných struktur a pravidel.



Obrázek 1 - schéma datových modelů  
(Vostrovský, 2014)

Stručný popis relačních databází nám nabízí tento popis: Databáze založená na relačním modelu, v němž jsou data logicky uspořádána do relací, tj. výsledků kartézského součinu nad doménami neboli množinami údajů. Fyzickou reprezentací relace je dvourozměrná tabulka, jejíž řádky obsahují kolekce údajů o sledovaném objektu (tzv. entity, n-tice, záznamy, věty) a ve sloupcích jsou vždy obsaženy údaje stejného typu (tzv. atributy, pole, položky). Tabulky a sloupce jsou identifikovány svými jmény, řádky jsou identifikovány primárním klíčem. Veškeré vztahy mezi daty jsou vyjádřeny rovněž pomocí

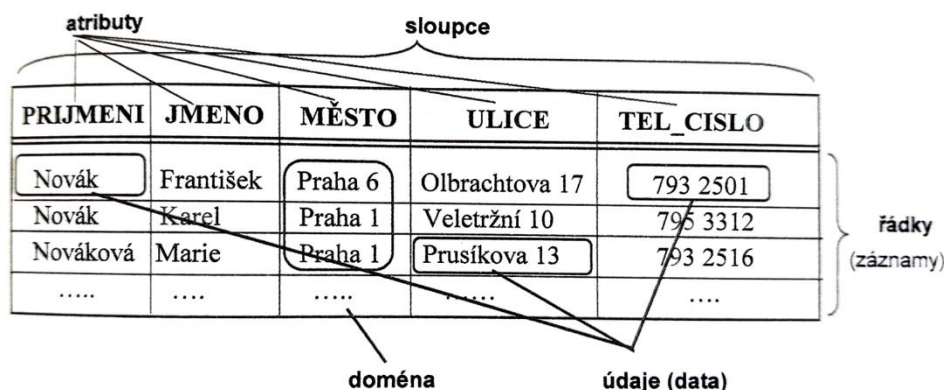
relací; vztahy mezi souvisejícími záznamy se definují prostřednictvím stejných údajů ve dvojici položek (cizí klíč). Množinový princip relačních databází umožňuje realizovat při manipulaci s daty neprocedurální operace založené na relační algebře a relačním kalkulu – projekce (výběr sloupců), selekce (výběr řádků), spojení (spojení řádků z různých tabulek na základě společné položky). (Pokorný, 1992)

Z popisu je jasné, jak relační databáze fungují. Jistě si zde zaslouží i zmínku soupeř a potenciální nástupce relačních databází, kterým se měly stát objektové databáze. Objektově orientované databáze jsou založeny na formálních prostředcích, které nemají přímou vazbu na architektury současných počítačů. Na pomyslné linii člověk-počítač stojí objektové databáze vlivem použitých prostředků blíže člověku. (Vostrovský, 2014)

### 1.1.1 Normalizace relační databáze

Při tvorbě relačních databází se musíme držet šesti normalizačních pravidel, která nám pomáhají k tomu, abychom rozumně navrhli tabulky a tím tak připravili strukturu, kterou budeme plnit daty a ty nám pak nadále zůstanou přehledná a logicky spojená.

Normální formy definuje požadavek na vlastnosti schématu relace z pohledu závislostí mezi atributy – hierarchie normálních forem (Codd: 1NF až 3NF, BCNF, 4NF, 5NF), tj. n-tá normální forma musí splňovat podmínky (n-1) normální formy a něco navíc. Požadavky na návrh založený na normalizaci: bezztrátovost dekompozice, zachování závislostí, dosažení minimálně BCNF, resp. 3NF. První normální forma (1NF) Relace je v první normální formě, právě když všechny její jednoduché domény obsahují pouze atomické hodnoty. Druhá normální forma (2NF) Relace je ve druhé normální formě, právě když je v 1NF a každý její neklíčový atribut, je plně funkčně závislý na každém kandidátním klíči. Třetí normální forma (3NF). Relace je ve třetí normální formě, právě když je ve 2NF a neexistuje žádný neklíčový atribut, který je tranzitivně závislý na některém kandidátním klíči. (Zendulka, 2003)



Obrázek 2 -  
Příklad relační  
tabulky

(Vostrovský,  
2014)

### 1.1.1.1 První normalizační forma

Jak již bylo zmíněno, tak první normalizační pravidlo nám říká, abychom zachovali pro každý atribut pouze atomické hodnoty. Příklad si můžeme ukázat v tabulce níže.

ID_Adresa	Systém_ID	Město	Ulice	Číslo popisné
1	1	Olomouc	Junácká	139
2	1	Plzeň	Pražská	1069
3	2	Dačice	Kolmá	256

Tabulka 1 - příklad 1NF

Pro nás to tedy znamená, že když chceme, aby sloupec obsahoval pouze atomické hodnoty, tak musíme například adresu rozdělit na jednotlivé části jako jsou název města, ulice a číslo popisné. Jak poznáme, že máme jednotlivé atributy správně rozděleny? Stačí se jednoduše zamyslet nad tím, zda když budu potřebovat upravit hodnotu v atributu nebo ji vyhledávat, takto půjde bez toho, aby daná změna měla dopad na význam celého řádek.

### 1.1.1.2 Druhá normalizační forma

Abychom se dostali k druhé normalizační formě, musíme splnit první normalizační formu. V druhé normalizační formě musíme zajistit, aby každý neklíčový atribut byl plně závislý na klíčovém. Za využití tabulky z 1NF a vytvořením nové tabulky systému si ukážeme, jak zajistit unikátnost a závislost na klíčových attributech. Kdybychom nevytvořili novou tabulku systémů, tak by se nám objevil Systém A 2x v tabulce Adres, což by ještě tolik nevadilo pro splnění podmínek 1NF, ale nevěděli bychom, jak zde svázat informace o systému na klíč tabulky Adres.

ID_systém	Nazev_systemu
1	Systém A
2	Systém B

Tabulka 2 - příklad 2NF

### 1.1.1.3 Třetí normalizační forma

Pro přistoupení k třetí normalizační formě je i zde nutné splnit první dvě normalizační formy. Třetí normalizační forma nám říká, že všechny neklíčové atributy musí být vzájemně nezávislé.

Co si pod touto definicí představit? To si můžeme ukázat na příkladu našich předešlých tabulek. Když si z těchto dvou tabulek vytvoříme novou a tu záměrně upravíme tak, aby nesplňovala třetí normalizační formu, vypadala by takto:

ID_Adresa	Systém_ID	Název systému	Město	Ulice	Číslo popisné
1	1	Systém A	Olomouc	Junácká	139
2	1	Systém A	Plzeň	Pražská	1069
3	2	Systém B	Dačice	Kolmá	256

Tabulka 3 - příklad 3NF

Máme tu tedy závislost, mezi neklíčovými atributy jako je název systému a zbylými atributy tabulky Adresa., protože ID\_adresa nám najednou určuje, z kterého systému daná adresa je.

### 1.1.2 Možnosti optimalizace

Možnosti optimalizace databáze a jejích výstupů je hned několik. Samozřejmě můžeme a měli bychom začít tou nejočividnější věcí a tou je správně navržená struktura databáze odpovídající normalizačním pravidlům. Také ovšem můžeme využít možnosti indexace dat.

Na základě Indexů lze velmi rychle (tj. nesequenčně) vyhledávat požadované věty z příslušné databázové tabulky. Index může rovněž sloužit i pro kontrolu jednoznačnosti pole (polí) v rámci celé tabulky. Vkládání duplicitního klíče (nebo změna klíče na již existující hodnotu klíče) jsou v případě požadavku na jednoznačnost zamítnuty. Klíč indexu je tvořen jedním nebo několika atributy tabulky, přičemž u každého atributu se specifikuje požadavek na vzestupné nebo sestupné třídění. Je ovšem třeba v této souvislosti poznamenat, že uživatel používání neovlivňuje, neboť volbu a použití indexů řídí databázový systém. Návrh indexů provádí správce databáze tak, aby předpokládané příkazy byly zpracovány co nejefektivněji. (Vostrovský, 2014)

Z popisu použití indexů je tedy zřejmé, že se jedná o další pomůcku, kterou využíváme již při návrh samotné databáze, ale hlavně při návrhu jednotlivých tabulek. Dále je ale nutné zmínit i nevýhody které se nám objeví, když začneme využívat indexace. Přes značné výhody indexů v souvislosti s rychlejším prováděním dotazů, lze poukázat i na některé nevýhody. Každý index musí být uložen, což znamená nárok na určitou část kapacity disku. Navíc je třeba indexy udržovat v aktuálním stavu, například při vložení nového záznamu či jeho vymazání je třeba provést i opravu indexu. Samotnou rychlost vyhledávání nelze příliš absolutizovat, neboť pokud vyhledáváme více jak 10 % záznamů z celkového

počtu obsažených v dotyčné databázové tabulce, stává se takovéto vyhledávání pomalejší než v případě neindexovaného. Nutno rovněž pamatovat na to, že při případném rušení dotyčné tabulky budou současně zrušeny i všechny její související indexy. (Vostrovský, 2014)

Dále se také vyplatí využít partition (oddíl). Tato vlastnost se dá využít současně s indexováním a tím zvýšit efektivitu v přístupu k datům. Dělení na oddíly je výkonná funkce, která umožňuje rozdělovat tabulky, indexy a tabulky uspořádané podle indexu na menší části, což umožňuje spravovat tyto databázové objekty a přistupovat k nim na jemnější úrovni granularity. Oracle poskytuje ucelenou řadu rozdělovacích schémat pro řešení každého obchodního požadavku. Navíc, protože je v příkazech SQL zcela transparentní, lze rozdělení na oddíly použít s jakoukoli aplikací, od balených aplikací OLTP po datové sklady.

Rozdělení:

Zvyšuje výkon tím, že pracuje pouze s daty, která jsou relevantní.

Zlepšuje dostupnost prostřednictvím správy jednotlivých oddílů.

Snižuje náklady ukládáním dat nejvhodnějším způsobem.

Snadno se implementuje, protože nevyžaduje žádné změny aplikací a dotazů.

Je vyzrálá, osvědčená funkce, kterou používají tisíce zákazníků Oracle.

### **Oracle podporuje širokou škálu metod dělení:**

Range Partitioning – data jsou distribuována na základě rozsahu hodnot rozdělení seznamu. Rozdělení dat je definováno samostatným seznamem hodnot. Jako klíč oddílu lze použít jeden nebo více sloupců.

Auto-List Partitioning – rozšiřuje možnosti metody seznamu automatickým definováním nových oddílů pro jakékoli nové hodnoty klíče oddílu.

Hash Partitioning – na rozdělovací klíč je použit interní hash algoritmus k určení oddílu.

Composite Partitioning – používá se kombinace dvou metod distribuce dat. Nejprve je tabulka rozdělena metodou distribuce dat jedna a poté je každá oblast dále rozdělena na pododdíly pomocí druhé metody distribuce dat.

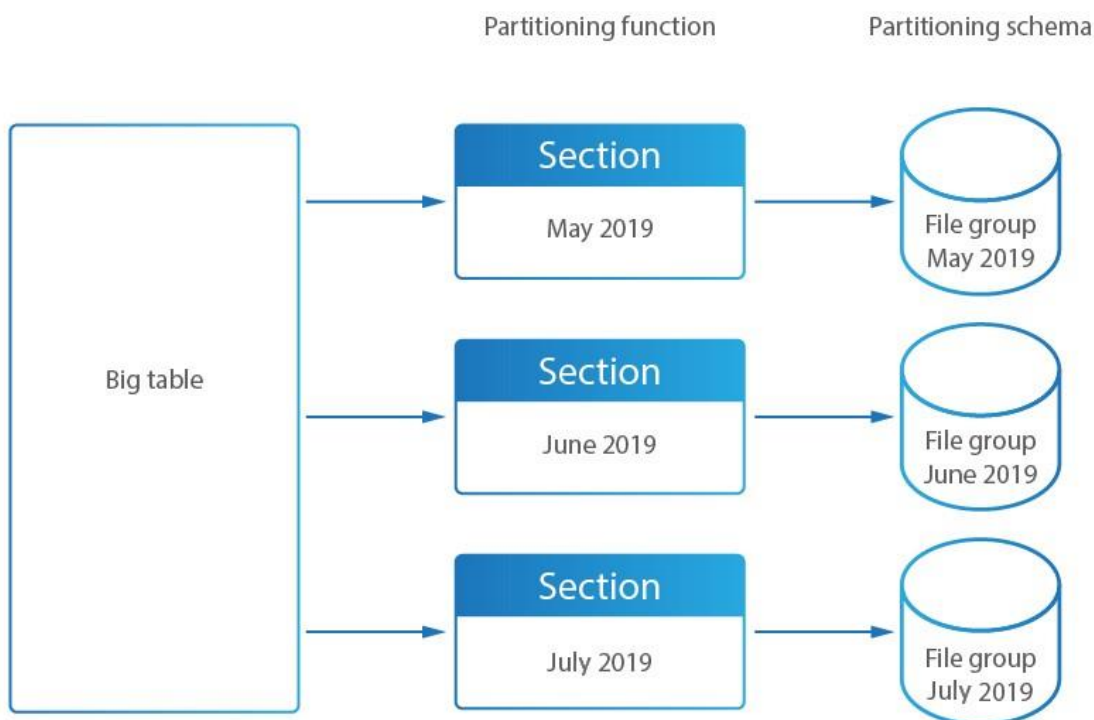
Multi-Column Range Partitioning – možnost, kdy se rozdělovací klíč skládá z několika sloupců a následující sloupce definují vyšší úroveň granularity než ty předchozí.

Interval Partitioning – rozšiřuje možnosti metody rozsahů automatickým definováním ekvivalentně rozdělených rozsahů pro jakékoliv budoucí oddíly pomocí definice intervalu jako součásti metadat tabulky.



Reference Partitioning Partitions – tabulka s využitím existujícího vztahu rodič-dítě. Vztah primárního klíče se používá ke zdědění strategie rozdělení nadřazené tabulky na podřízenou tabulku.

Virtuální dělení podle sloupců – umožňuje, aby rozdělovacím klíčem byl výraz, který používá jeden nebo více existujících sloupců tabulky a ukládá výraz pouze jako metadata. Interval Reference Partitioning – rozšíření k dělení na oddíly, které umožňuje použití intervalově dělených tabulek jako nadřazených tabulek pro dělení referencí. (Oracle, 2022)



Obrázek 3 - schéma partitioning (DataSunrise, 2022)

## 1.2 ERD model

Při návrhu relační databáze je nejdůležitější si uvědomit, jak vlastně naše databáze bude vypadat a jak budou vypadat vazby mezi jednotlivými tabulkami. Pomůže nám to nejen pro lepší představu, jak naše tabulky budou vypadat, ale hlavně nám to umožní už si předem uvědomit případné chyby, které mohou vzniknout při prvotních myšlenkách.

Diagram vztahů entit (ERD), známý také jako model vztahů entit, je grafické znázornění zobrazující vztahy mezi osobami, objekty, místy, koncepty nebo událostmi v

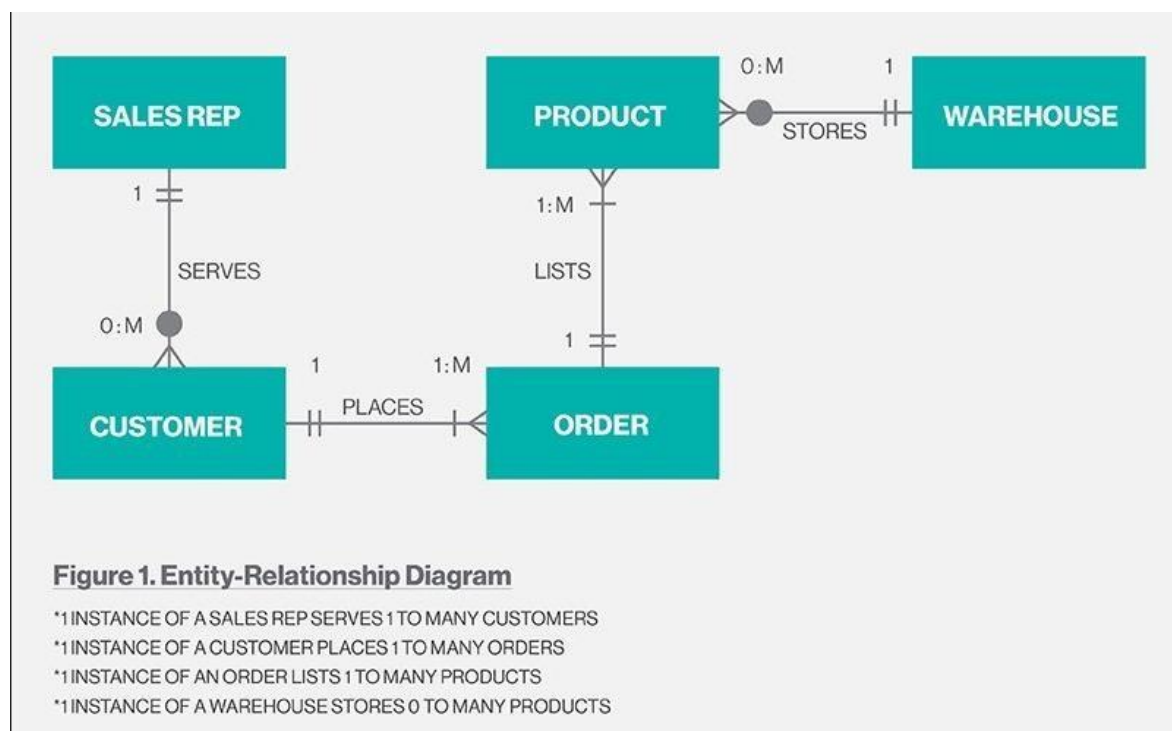
rámci systému informačních technologií (IT). ERD používá techniky datového modelování, které mohou pomoci definovat obchodní procesy a sloužit jako základ relační databáze. Diagramy vztahů entit představují vizuální výchozí bod pro návrh databáze, jenž lze rovněž použít k určení požadavků na informační systém v celé organizaci. I po zavedení relační databáze může ERD sloužit jako referenční bod, pokud by bylo později nutné provést ladění nebo změnu obchodních procesů.

Zatímco však ERD může být užitečná pro organizaci dat, která lze reprezentovat relační strukturou, nemůže dostatečně reprezentovat polostrukturovaná nebo nestrukturovaná data. Je také nepravděpodobné, že by byla sama o sobě užitečná při integraci dat do již existujícího informačního systému. Tři hlavní kardinální charakteristiky jsou:

Vztah jedna ku jedné (1:1). Pokud je například každý zákazník v databázi spojen s jednou poštovní adresou.

Vztah jeden k mnoha (1:M). Například jeden zákazník může zadat objednávku na více výrobků. Zákazník je spojen s více entitami, ale všechny tyto entity mají jediné spojení zpět k témuž zákazníkovi.

Vztah mnoho k mnoha (M:N). Například ve společnosti, kde všichni agenti call centra pracují s více zákazníky, je každý agent spojen s více zákazníky a více zákazníků může být také spojeno s více agenty. (Biscobing, 2023)



Obrázek 4 - Příklad ERD (Biscobing, 2023)

Pokud si tedy dáme opravdu tu práci a připravíme kompletní návrh databáze a vztahů, tak odhalíme nedostatky, které náš návrh již obsahuje. Toto nám ušetří spoustu času, práce a peněz při samotné implementaci, nebo nám naopak může dojít, že nám i nějaké zásadní tabulky z našeho návrhu schází a je třeba celý návrh rozšířit. Je totiž běžné, že nám spoustu věcí může vypadnout, dokud nevidíme celý problém popsany na jednom místě.

### **1.3 Reportování dat**

Pokud se zajímáme o to, jak databáze pracuje, tak musíme brát v potaz i jak a za jakou dobu má uživatel data dostupná, protože uživatel má prospěch z databáze jen pokud je schopen získat data a ta dále zpracovávat.

Data uložená v sebelepší databázi nám nebudou moc nápomocná, pokud je nebudeme schopni včas a správně zobrazit a dále zpracovat. Zpracování dat může být různé ať už pro zobrazení v libovolné aplikaci, kde se vyplní jednotlivá pole, tak pro potřeby reportů, kde se data mohou objevovat stále jako tabulky a mnohem důležitěji dále jako grafy.

K těmto potřebám máme nepřeberné množství možností v podobě různých aplikací, kdy tyto aplikace slouží jako prostředek k přehlednému, agregovanému a transformovanému zobrazení dat uložených v databázi.

Samotné propojení databáze s touto aplikací má dva rozměry. Těmi jsou síťová dostupnost a pak samotný optimalizovaný dotaz SQL, díky kterému určíme, jaká data chceme získat z databáze a samotné zpracování dotazu. V případě, že řešíme, jak dlouho bude trvat data zobrazit, musíme brát v potaz oba tyto faktory.

#### **1.3.1 Co je data reporting?**

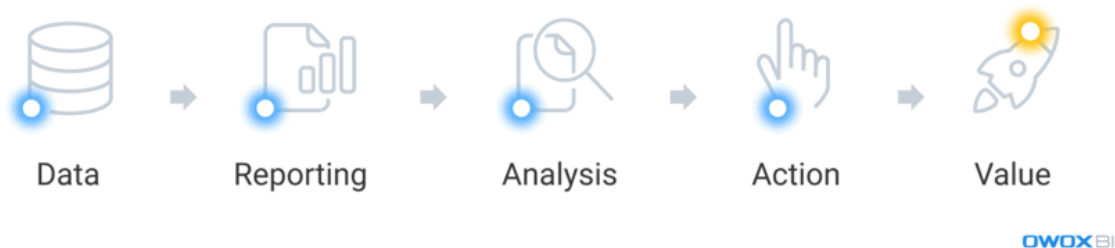
Reportování dat vám pomůže sledovat, co se děje s vaší firmou, a vyhodnocovat její výkonnost. Je to proces shromažďování, slučování a vizualizace nezpracovaných dat ze všech dostupných zdrojů. Nejčastěji se prezentují ve formě tabulek, grafů nebo diagramů.

Nezapomínejte na tyto dvě věci:

Ve většině případů evidence založená na datech zobrazuje pouze historická data, takže vidíte vyhodnocení minulých akcí, které již nemůžete změnit.

Chcete-li adekvátně vyhodnotit průběžnou výkonnost své značky, musíte pochopit kontext (přínejmenším zvážit odvětví a obor, ve které vaše společnost působí).

(Malysheva, 2023)



Obrázek 5 - Kroky pro report (Malysheva, 2023)

### 1.3.2 Jak vytvářet datové reporty

Analýzou dat můžete činit informovaná rozhodnutí a testovat pracovní hypotézy. Můžete upřesnit, kam směřují zdroje vaší společnosti, jakého pokroku bylo dosaženo a čemu by vaše společnost měla věnovat největší pozornost.

Chcete vědět, jak napsat zprávu o analýze dat? Podívejme se na kroky, které musíte udělat, abyste vytvořili to, co dokonale vyhovuje vaší společnosti:

Krok 1. Určete účel zprávy a konkrétní otázky, na které by měla odpovídat. Různí specialisté potřebují různé reporty a samozřejmě i odpovědi na různé otázky. A příliš mnoho otázek v jednom přehledu jej může přetížit.

Krok 2. Definujte metriky a zdroje dat. Jakmile se rozhodnete pro otázky, na které chcete odpovídat, a pro specialisty, kteří budou dashboard používat, měli byste zvýraznit kritické metriky. Také byste měli určit, jaké informace jsou potřeba k sestavení reportů a jaké zdroje by měly být k těmto reportům připojeny.

Krok 3. Ujistěte se, že sběr dat funguje správně. Abyste mohli přijímat informovaná rozhodnutí, musíte si být jistí kvalitou svých dat. Ujistěte se, že informace jsou shromažďovány přesně a bez chyb. Mějte také na paměti, že váš atribuční model by měl být přizpůsoben vašim obchodním potřebám. (Malysheva, 2023)



Obrázek 6 - Rady pro tvorbu reportu (Malysheva, 2023)

### 1.3.3 Jak správně zobrazit data

Pokud jsme schopni zobrazit data pro report v rozumném čase, nastává další hlavní krok a tím je správná interpretace dat a jejich příprava pro koncové uživatele. Uživatelé by měli dostat správnou a námi chtěnou informaci.

Toto je poslední a nejdůležitější krok k tomu, jak správně předávat data a informace z nich získaná koncovým uživatelům. Z toho důvodu se pak u grafů a tabulek využívá určitého barevného schématu. Komplementární schéma: kombinací doplňkových (komplementárních) barev, docílíme harmonického kontrastu: barvy vedle sebe vyniknou, vzájemně se podpoří a jedna druhou nepřehluší. analogické schéma: – barvy ležící v kruhu vedle sebe, vykazují minimální kontrast. Split – komplementární schéma: kombinace jedné barvy z analogickými barvami a její komplementární barvou. (Vávrová, 2022)



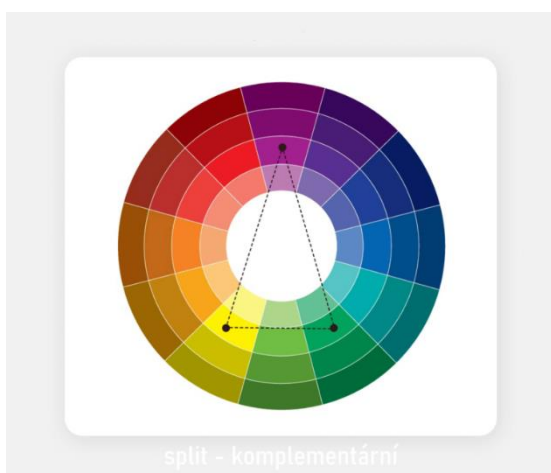
komplementární barvy

Obrázek 7 - komplementární barvy (Vávrová, 2022)



analogické barvy

Obrázek 8 - analogické barvy (Vávrová, 2022)



split - komplementární

Obrázek 9 - split – komplementární (Vávrová, 2022)

### **3. Vlastní práce**

V této části se primárně zaměříme na návrh databáze. Tento návrh bude sloužit k uložení klientských dat. Databázi pak uživatelé budou moci využívat jako základ pro tvorbu reportů a datovou analýzu. Díky tomuto návrhu bude pro uživatele snadnější zpracovávat data.

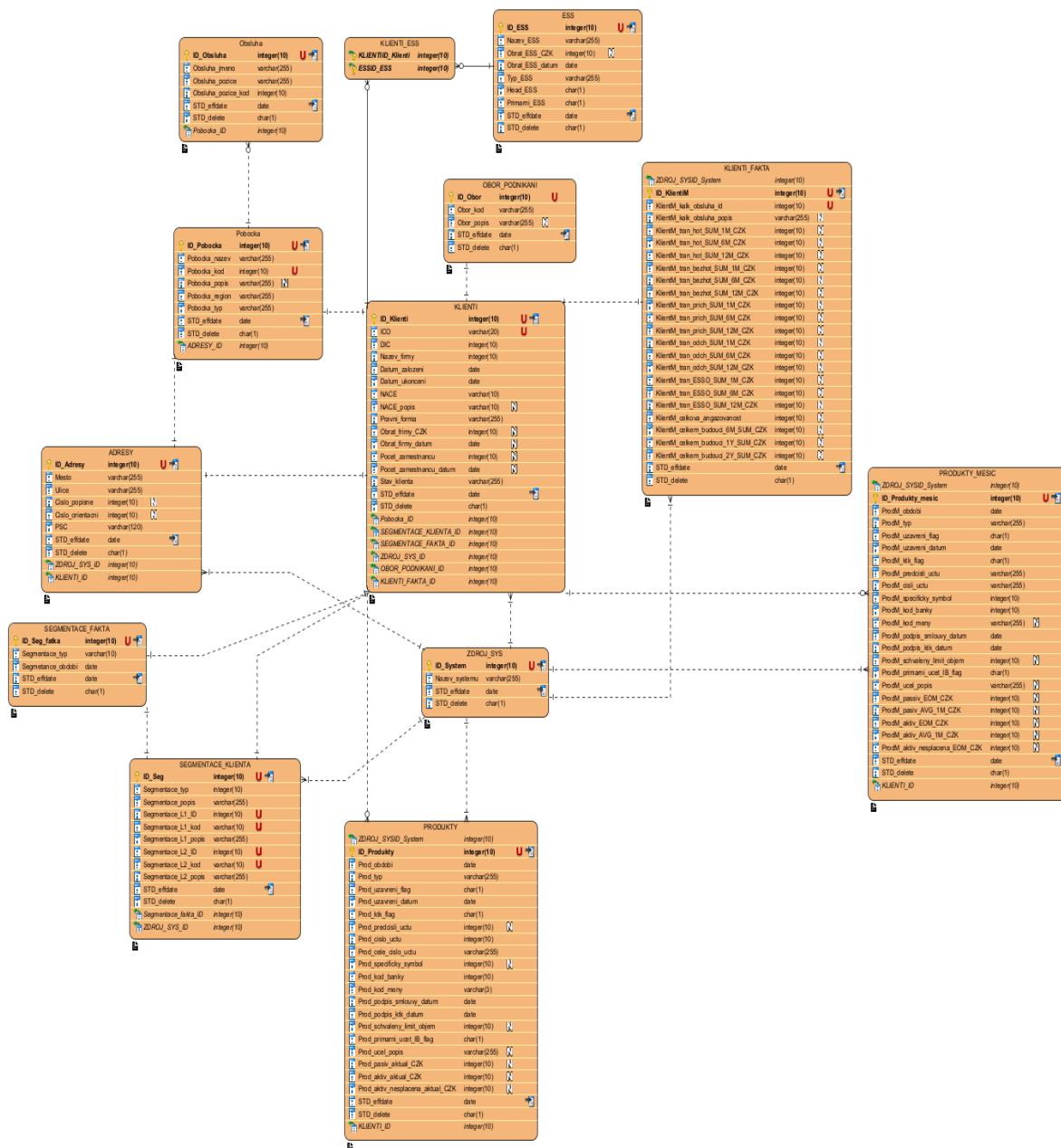
Celý návrh bude tvořen pomocí entity relationship modelu. Tabulky zde budou popsány jako jednotlivé bloky s atributy a budou mezi nimi vazby. V rámci jednotlivých vazeb je i znázorněna kardinalita. Návrh je tvořen podle odpovídajících normalizačních forem do třetí úrovně.

#### **1.4 Popsání datového modelu**

Naším pomyslným středem v tomto datovém modelu je tabulka s údaji o klientech, na kterou budeme dále dle potřeb napojovat ostatní tabulky. Tímto způsobem nejlépe zajistíme modularitu modelu. Díky tomuto přístupu lze tvořit na sobě nezávislé reporty a zároveň zajistit konzistenci mezi reporty ve smyslu stejné datové základny pro všechny reporty.

Další tabulky pak budou tvořeny dle normálních forem. Data budou takto rozdělena do logických celků a tyto celky si dále podrobněji rozebereme. Každá tabulka obsahuje technické sloupce, které udávají za kdy byl záznam přidán a zda není záznam neplatný.

Všechny informace do databáze jsou získávány z veřejných zdrojů, případně přímo od klientů.



Obrázek 10 - ERD návrh

### 1.4.1 Tabulka Klienti

Naše hlavní tabulka, od které se bude odvíjet spousta našich reportů. V našem podání je klient základním stavebním kamenem, protože tato tabulka obsahuje informace a identifikátory klientů. Bez této tabulky nejsme schopni rozeznat o kterého klienta se jedná, a proto je pro nás důležitá.

Kromě identifikačních údajů klienta tato tabulka dále obsahuje detailnější data o klientech jako jsou právní forma klienta, obor podnikání, datum založení firmy,



případné datum, kdy byla firma ukončena, případně pro analytické potřeby data o počtu zaměstnanců a firemního obratu.

Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Nullabl e	Unique
KLIENTI	Tabulka obsahující základní popisné informace o klientech.	Datum_L koncen		date	0	false	false	false
		Datum_z alozeni		date	0	false	false	false
		DIC		integer	10	false	false	false
		ICO		varchar	20	false	false	true
		ID_Klient		integer	10	true	false	true
		KLIENTI_FAKTA_ID		integer	10	false	false	false
		NACE		varchar	10	false	false	false
		NACE_po pis		varchar	10	false	true	false
		Nazev_f rmy		integer	10	false	false	false
		OBOR_F ODNIKA NI_ID		integer	10	false	false	false
		Obrat_fir my_datu m		date	0	false	true	false
		Obrat_fri my_CZK		integer	10	false	true	false
		Pobocka _ID		integer	10	false	false	false
		Pocet_zr mestnan cu		integer	10	false	true	false
		Pocet_zr mestnan ou_datu m		date	0	false	true	false
		Pravni_f orma		varchar	255	false	false	false
		SEGMIEN TACE_FA KTA_ID		integer	10	false	false	false
		SEGMIEN TACE_KL IENTA_I D		integer	10	false	false	false
		Stav_klie nta		varchar	255	false	false	false
		STD_del ete		char	1	false	false	false
		STD_effd ate		date	0	false	false	false
		ZDROJ_S YS_ID		integer	10	false	false	false

Obrázek 11 - metadata tabulky klienti

## 1.4.2 Tabulka ESS

Zkratka ESS znamená Ekonomicky spjaté skupiny. Ekonomicky spjatou skupinou je myšlena skupina firem a osob, které jsou nějakým způsobem v obchodním spojení s jinou firmou nebo osobou.

Data v této tabulce slouží tedy pouze k tomu, aby popsala jednotlivé ESS, jejich vazby případně již předpřipravené agregované obraty. Zároveň zde máme i atribut pro rozpoznání typu ESS, protože ESS můžeme dle různých pohledů a pravidel chápat jinak. Proto je důležité vždy vědět, o které ESS mluvíme. Jednou můžeme do ESS zahrnout i osoby spojené s firmami, nebo jen firmy. Zároveň je také nutné určit primární ESS, tedy tu, ve které tento klient figuruje z regulačních důvodů. Dále je důležité vědět, kde je daný klient hlavním figurantem, myšleno tzv. headem.

Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique
ESS	Tabulka obsahující základní informace o ESS	Head_ESS		char	1	false	false	false
		ID_ESS		integer	10	true	false	true
		Nazev_ESS		varchar	255	false	false	false
		Obrat_ESS_CZK		integer	10	false	true	false
		Obrat_ESS_datum		date	0	false	false	false
		Primarni_ESS		char	1	false	false	false
		STD_delete		char	1	false	false	false
		STD_effc ate		date	0	false	false	false
		Typ_ESS		varchar	255	false	false	false

Obrázek 12 - metadata tabulky ESS

## 1.4.3 Tabulka Produkty

V této tabulce se nachází data o produktech klientů. Jedná se zejména o podrobný popis produktů a jejich nastavení parametrů.

Data jako jsou období trvání produktu, typ produktu, příznak o uzavření produktu, případné datum uzavření, které bude vyplněno datem uzavření, případně konstantou datumu 1.1.1900, pokud nedošlo ještě k uzavření produktu. Dále je tu příznak o tom, zda je k běžnému účtu vázán kontokorent.

Samozřejmě se zde nachází i důležité identifikátory jako jsou číslo účtu, případně jeho předčíslí, specifický symbol a kód banky.

Pro rychlejší zpracovávání dat v analýze je zde i atribut, který tyto atributy spojuje do jednoho jako celé číslo účtu. V tabulce najdeme také informace o měně produktů a podpisech jednotlivých smluv. K tomu nám slouží tyto atributy jako jsou datum podpisu smlouvy, datum podpisu smlouvy kontokorentu, dále data o schválené výši limitu klienta. Nalezneme tu i označení pro primární účet, ke kterému je vedena internet banka, data o účelu produktu, to se týká primárně úvěrů a jiných podobných specifických produktů vyžadující bližší popis. V neposlední řadě jsou zde informace o aktuálním zůstatku pasiv, aktiv a nesplacené částce.

Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique
PRODUKTY	Data v tabulce obsahují informace o produktech v jejich aktuálním stavu v systémech.	ID_Produkty		integer	10	true	false	true
		KLIENTI_ID		integer	10	false	false	false
		Prod_aktiv_aktual_CZK		integer	10	false	true	false
		Prod_aktiv_nesplacena_aktual_CZK		integer	10	false	true	false
		Prod_cel_cislo_uctu		varchar	255	false	false	false
		Prod_cislo_uctu		integer	10	false	false	false
		Prod_kod_banky		integer	10	false	false	false
		Prod_kod_meny		varchar	3	false	false	false
		Prod_ktk_flag		char	1	false	false	false
		Prod_obdobu		date	0	false	false	false
		Prod_pasiv_aktual_CZK		integer	10	false	true	false
		Prod_podpis_ktk_datum		date	0	false	false	false
		Prod_podpis_smlouvy_datum		date	0	false	false	false
		Prod_predisli_uctu		integer	10	false	true	false
		Prod_pri_marni_uct_IB_flag		char	1	false	false	false
		Prod_schvaleny_jimik_objem		integer	10	false	true	false
		Prod_specificky_symbol		integer	10	false	true	false
		Prod_typ		varchar	255	false	false	false
		Prod_ucel_popis		varchar	255	false	true	false
		Prod_uzavreni_datum		date	0	false	false	false
		Prod_uzavreni_flag		char	1	false	false	false
		STD_dete		char	1	false	false	false
		STD_effdate		date	0	false	false	false
		ZDROJ_SYSID_System		integer	10	false	false	false

Obrázek 13-  
metadata tabulky  
Produkty

#### 1.4.4 Tabulka Produkty\_mesic

Produkty\_mesic je tabulka obsahující agregované informace po měsících o produktech. Přepočítání této tabulky je daný k fixnímu datu, zpravidla ke konci daného měsíce. Agregovaná a unifikovaná data pomohou k vyšší efektivitě při vytváření a sledování trendů. U produktů chceme hlavně sledovat pasivní a aktivní zůstatky a díky měsíčnímu odstupu při napočítávání dat tak můžeme sledovat i průměrné zůstatky. Zároveň se v tabulce objeví i atributy již obsažené v tabulce Produkty, atributy jako jsou například predcislí\_uctu, cislo\_uctu, specificky\_symbol a další.

Tyto atributy jsou obsaženy v obou tabulkách z důvodu jejich nezávislého využití. Sice si zde tyto atributy duplikujeme, ale tím pádem můžeme nezávisle využívat obě tyto tabulky a nebude potřeba jejich neustálé propojování.

Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique
PRODUKTY_MESIC	Data jsou zaměřena na produkty kdy se v této tabulce agregují po měsících pro možnost reportování.	ID_Produkty_mesic		integer	10	true	false	true
		KLIENTI_ID		integer	10	false	false	false
		ProdM_aktiv_AVG_IM_CZK		integer	10	false	true	false
		ProdM_aktiv_EOM_CZK		integer	10	false	true	false
		ProdM_aktivnes_placena_EOM_CZK		integer	10	false	true	false
		ProdM_cislo_uctu		varchar	255	false	false	false
		ProdM_kod_banky		integer	10	false	false	false
		ProdM_kod_meny		varchar	255	false	true	false
		ProdM_ktk_flag		char	1	false	false	false
		ProdM_období		date	0	false	false	false
		ProdM_pasiv_AVG_IM_CZK		integer	10	false	true	false
		ProdM_pasiv_EOM_CZK		integer	10	false	true	false
		ProdM_podpis_ktk_datum		date	0	false	false	false
		ProdM_podpis_smlouvy_datum		date	0	false	false	false
		ProdM_podpis_redcislo_uctu		varchar	255	false	false	false
		ProdM_primarni_uct_IB_flag		char	1	false	false	false
		ProdM_schvaleny_limit_objem		integer	10	false	true	false
		ProdM_specificky_symbol		integer	10	false	false	false
		ProdM_uzivatel_yp		varchar	255	false	true	false
		ProdM_uzivatel_popis		varchar	255	false	true	false
		ProdM_uzivatel_zavreni_datum		date	0	false	false	false
		ProdM_uzivatel_zavreni_flag		char	1	false	false	false
		ProdM_uzivatel_zavreni_lag		char	1	false	false	false
		STD_dle		char	1	false	false	false
		STD_effektivita		date	0	false	false	false
		ZDROJ_SYSDYSTEM		integer	10	false	false	false

Obrázek 14- metadata tabulky Produkty\_mesic

### 1.4.5 Tabulka Pobočka

Hlavním cílem tabulky Pobočka je nás informovat, t na které pobočce je klient obsluhován, případně kde je domovská pobočka bankéře. Tato data jsou důležitá pro přehled řízení aktivit a efektivity obsluhy.

Zároveň lze podle typu pobočky dohledat informace o tom, jak je pobočka vybavená, případně zda její provoz dokáže pokrýt všechny potřeby zákazníků. Jsou jisté věci jako například vklad hotovosti přes vkladomat. Tato služba není potřeba, aby byla dostupná na všech pobočkách, ale třeba jen v nějakém okruhu. Toto můžeme aplikovat na více parametrů a ve spojení s tabulkou Adresy může takto optimalizovat náklady na provoz jednotlivých poboček.

Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique
Pobočka	Tabulka která obsahuje informace o fyzických pobočkách a zároveň doplňuje informace o obsluze.	ADRESY_ID		integer	10	false	false	false
		ID_Pobočka		integer	10	true	false	true
		Pobočka_kod		integer	10	false	false	true
		Pobočka_nazev		varchar	255	false	false	false
		Pobočka_popis		varchar	255	false	true	false
		Pobočka_region		varchar	255	false	false	false
		Pobočka_typ		varchar	255	false	false	false
		STD_delete		char	1	false	false	false
		STD_effdate		date	0	false	false	false

Obrázek 15- metadata tabulky Pobočka

#### 1.4.6 Tabulka Klienti\_fakta

Tabulka Klienti\_fakta je tvořena podobně jako tabulka Porodky\_mesic. Jde o agregovaný pohled na klienta a jeho transakce. Je zde navíc ještě agregace po 6 a 12 měsících.

Zároveň zde díky této agregaci historie můžeme využít data pro natrénování statistického modelu a dále pro zpracování pomocí statistických nástrojů a datascience využijeme pro predikci budoucích pohybů na účtech. Užitečné je to hned v několika směrech.

Při znalosti toho, jak klient pracuje se svými zůstatky, můžeme odhalit potenciál k obchodu. Dále nám tato informace poslouží při zpracování klientů, kteří prochází procesem schvalování nových produktů. Budeme vědět, jak si klient vede a nejspíše i povede a budeme schopni zjistit, co si klient může dovolit a tím mu i přizpůsobit nabídky. Naopak, pokud klientova situace bude mít tendenci jít spíše negativním směrem, můžeme včas klientovi pomoci k tomu, aby nezašla až příliš daleko.

Díky rozlišení dat na transakce hotovostní a bezhotovostní můžeme sledovat i klientovi potřeby. A znovu se lze tímto směrem vrátit k příkladu z popisu tabulky Pobočka. Pokud víme, jak moc klienti využívají nebo nevyžívají hotovost, můžeme dále optimalizovat chod poboček a s tím i spojené náklady, případně i podle toho upravit nabídku produktů.



Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique	Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique	
KLIENTI_FAKTA	Data zde obsažená jsou agregovaná pourčítém časovém období a jsou navázána na daného klienta.	ID_Klientim		integer	10	true	false	true			KlientM_tran_ESSO_SUM_6M_CZK		integer	10	false	true	false	
		KlientM_celkem_budouci_1Y_SUM_CZK		integer	10	false	true	false				KlientM_tran_hot_SUM_12M_CZK		integer	10	false	true	false
		KlientM_celkem_budouci_2Y_SUM_CZK		integer	10	false	true	false				KlientM_tran_hot_SUM_1M_CZK		integer	10	false	true	false
		KlientM_celkem_budouci_6M_SUM_CZK		integer	10	false	true	false				KlientM_tran_hot_SUM_6M_CZK		integer	10	false	true	false
		KlientM_celkova_angazovanost		integer	10	false	true	false				KlientM_tran_odch_SUM_12M_CZK		integer	10	false	true	false
		KlientM_kalk_obs_luha_id		varchar	255	false	true	false				KlientM_tran_odch_SUM_1M_CZK		integer	10	false	true	false
		KlientM_kalk_obs_luha_popis		integer	10	false	true	false				KlientM_tran_odch_SUM_6M_CZK		integer	10	false	true	false
		KlientM_tran_bez_hot_SUM_12M_CZK		integer	10	false	true	false				KlientM_tran_prich_SUM_12M_CZK		integer	10	false	true	false
		KlientM_tran_bez_hot_SUM_1M_CZK		integer	10	false	true	false				KlientM_tran_prich_SUM_1M_CZK		integer	10	false	true	false
		KlientM_tran_bez_hot_SUM_6M_CZK		integer	10	false	true	false				KlientM_tran_prich_SUM_6M_CZK		integer	10	false	true	false
		KlientM_tran_ESSO_SUM_12M_CZK		integer	10	false	true	false				STD_del_ete		char	1	false	false	false
		KlientM_tran_ESSO_SUM_1M_CZK		integer	10	false	true	false				STD_effdate		date	0	false	false	false
												ZDROJ_SYSD_System		integer	10	false	false	false

Obrázek 16- metadata tabulky Klienti\_fakta

### 1.4.7 Tabulka Adresy

Tabulka adresy obsahuje kontaktní informace na klienty a data o tom, kde jednotlivé firmy dle rejstříků sídlí. Tato tabulka je zároveň důležitá a doplňková část pro tabulku Pobočka. V tabulce také nalezneme atribut System\_ID, jelikož adresy klientů, poboček a jiných systémů je potřeba jednotlivé adresy rozlišit. Protože zároveň každá pobočka musí mít svou adresu.

V tabulce jsou jednotlivé adresy rozloženy do několika atributů, tak aby byla zachována jejich elementárnost a dle potřeba se dalo poskládat jednotlivé adresy.

Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique
ADRESY	Data zde jsou uloženy adresy poboček a sídel klientů.	Cislo_orientacni		integer	10	false	true	false
		Cislo_pospisne		integer	10	false	true	false
		ID_Adresy		integer	10	true	false	true
		KLIENTI_ID		integer	10	false	false	false
		Mesto		varchar	255	false	false	false
		PSC		varchar	120	false	false	false
		STD_delete		char	1	false	false	false
		STD_effc		date	0	false	false	false
		Ulice		varchar	255	false	false	false
		ZDROJ_YS_ID		integer	10	false	false	false

Obrázek 17 - metadata tabulky Adresy

#### 1.4.8 Tabulka Obsluha

Informace o obsluze nám pomáhají při práci s obslužnými modely a zároveň si také můžeme hlídat správnou obsluhu klientů tedy zda opravdu mají přiřazenou obsluhu a zároveň zda je přiřazená obsluha správná. Je důležité, aby každý klient měl přiřazenou obsluhu a tím o něj bylo postaráno.

Tabulka Obsluha obsahuje primárně informace o bankéřích a zaměstnancích, tzn. kdo a na jaké pozici pracuje. Tyto informace jsou důležité pro správný chod poboček a obsluhu klientů.

Tabulka Obsluhy sama o sobě není až tak všeříkající a je třeba tuto tabulku spojit primárně s tabulkou Pobočka a případně s tabulkou Adresy.

Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique
Obsluha	Tato tabulka obsahuje informace o obluze klientů, primárně zaměřeno na zaměstnance	ID_Obsluha		integer	10	true	false	true
		Obsluha_jmeno		varchar	255	false	false	false
		Obsluha_pozice		varchar	255	false	false	false
		Obsluha_pozice_kod		integer	10	false	false	false
		Pobočka_ID		integer	10	false	false	false
		STD_delete		char	1	false	false	false
		STD_effdate		date	0	false	false	false

Obrázek 18 - metadata tabulky Obsluha

### 1.4.9 Tabulka Zdroj\_sys

Tato tabulka je spíše číselníkem, kdy její primární úkol je převádět ID jednotlivých systémů na jejich slovní název nebo popis. Přesto je tato tabulka velmi důležitá, protože jednotlivá data se k nám dostávají z různých zdrojů, a to může vyvolávat mnoho rozporů.

Může nastat následující situace: uvidíme adresu, která může být stejná jak pro klienta, tak třeba pro naši pobočku. Ale každá z těchto adres bude pocházet z jiného zdrojového systému. Díky této tabulce je poté můžeme rozeznat a správně s nimi pracovat.

Pokud máme data z vícero zdrojů, je nejvhodnějším způsobem data filtrovat právě podle jejich zdrojového systému, můžeme tím rychle provádět analýzu nebo úpravy nad daty z různých systémů.

Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique
ZDROJ_SYS	V tato tabulka slouží spíše jako číselník pro zdrojové systémy, aby se snadno dalo dohledat, které ID je který zdrojový systém.	ID_System		integer	10	true	false	true
		Nazev_systemu		varchar	255	false	false	false
		STD_delete		char	1	false	false	false
		STD_effdate		date	0	false	false	false














Obrázek 19- metadata tabulky Zdroj\_sys

#### 1.4.10 Tabulka Segmentace\_klienta

Segment klienta je alfou a omegou pro práci s klientem. Pokud máme klienta správně obsluhovat, je potřeba ho správně zařadit do segmentu. Segment u klienta ovlivňuje hned několik parametrů.

Tím pádem je velké plus, pokud máme někde vypsané, v jakém segmentu se klient přímo pohybuje. Žádný uživatel ani manažer nechce pořád dokola kontrolovat, zda daný klient splňuje kombinaci různých podmínek, aby věděl, do kterého segmentu daný klient spadá.

V tabulce se také objeví atributy se sufixem L1 a L2. Jedná se o atributy rozlišující „vrstvu“ nebo také detail dané informace. Segmenty klientů se totiž rozpadají ještě na další detailnější sub segmenty.

Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique
SEGMENTACE_KLIENTA	Segmentace klienta obsahuje data o segmentu klienta a jeho sub segmentu.	 ID_Seg		integer	10	true	false	true
		 Segmentace_fakta_ID		integer	10	false	false	false
		 Segmentace_L1_ID		integer	10	false	false	true
		 Segmentace_L1_kod		varchar	10	false	false	true
		 Segmentace_L1_popis		varchar	255	false	false	false
		 Segmentace_L2_ID		integer	10	false	false	true
		 Segmentace_L2_kod		varchar	10	false	false	true
		 Segmentace_L2_popis		varchar	255	false	false	false
		 Segmentace_popis		varchar	255	false	false	false
		 Segmentace_typ		integer	10	false	false	false
		 STD_delete		char	1	false	false	false
		 STD_effdate		date	0	false	false	false
		 ZDROJ_SYS_ID		integer	10	false	false	false

Obrázek 20 - metadata tabulky Segmentace\_klienta

### 1.4.11 Tabulka Segmentace\_fakta

Tabulka Segmentace\_fakta obsahuje data, která se vztahují k segmentaci. V této tabulce se shromažďují data v agregované podobě tak, aby se s nimi dalo pracovat v reportech. Tabulky agregující data třeba v měsíčních přehledech jsou výhodné z toho důvodu, že nám krásně schovají krátkodobé změny, ke kterým dochází.

Pokud bychom využívali pouze tabulku Segmentace\_klienta, tak bychom museli složitě vybírat třeba jednotlivé dny, abychom si vytvořili novou časovou osu.

Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique
SEGMENTACE_FAKTA	Agregovaná data o segmentu klienta, uchovávána po měsících.	ID_Seg_fakta		integer	10	true	false	true
		Segmentace_typ		varchar	10	false	false	false
		Segmentace_obdobli		date	0	false	false	false
		STD_delete		char	1	false	false	false
		STD_effdate		date	0	false	false	false

Obrázek 21 - metadata tabulky Segmentace\_fakta

### 1.4.12 Tabulka Obor\_podnikani

Tabulka Obor\_podnikani je číselníkem oborů podnikání klientů. Tato tabulka slouží jako překladová tabulka pro ID obsažená v tabulce Klienti a dodání jeho popisu.

Tato data jsou oddělena v nezávislé tabulce z důvodu možnosti jednotlivých úprav a možnosti nezávislého využití pro report oborů podnikání, kde není nutná vazba na klienta.

Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique
OBOR_PODNIKANI	Data v této tabulce obsahují informace o oborech podnikání klientů.	ID_Obor		integer	10	true	false	true
		Obor_kod		varchar	255	false	false	false
		Obor_popis		varchar	255	false	true	false
		STD_delete		char	1	false	false	false
		STD_effdate		date	0	false	false	false

Obrázek 22 - metadata tabulky Obor\_podnikani

## Výsledky a diskuse

Naším hlavním tématem k diskusi bude časová náročnost práce s databází oproti času zpracování pomocí uživatelů. Protože tyto dva světy mají různý přístup k tomu, jakou formou a způsobem by chtěly data dostávat.

Z jedné strany je totiž požadováno, aby data v databázi byla snadno přístupná a pro běžného uživatele srozumitelná. Tento požadavek nelze ale vždy splnit, a to hlavně v případech, pokud bereme v potaz normalizační formy.

### 1.5 Původní návrh databáze

Nyní se pojdme podívat na to, jak vlastně vypadá naše databáze nyní a popsat hlavní rozdíly, které naším novým návrhem vznikly.

Hlavním bodem změn je samotná tabulka Klienti, která v původním návrhu obsahuje jak přímo informace o klientech, tak i konkrétní informace o službě a pobočkách, včetně adres. Toto řešení ale není dlouhodobě únosné jak pro práci uživatelů s daty, tak ani pro práci databáze jako takové.

Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique	Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique		
KLIENTI	Tabulka obsahující základní popisné informace o klientech.	Bankastav_id		integer	10	false	false	false			Pocet_zaměstnancu_datum		date	0	false	true	false		
		Bankastav_popis		integer	10	false	false	false			Pravni_forma		varchar	255	false	false	false		
		Datum_končení		date	0	false	false	false			SEGMENTACE_FAKTA_ID		integer	10	false	false	false		
		Datum_založení		date	0	false	false	false			SEGMENTACE_KLIENTA_ID		integer	10	false	false	false		
		Dcerastav_id		integer	10	false	false	false			Skupinastav_id		varchar	255	false	false	false		
		Dcerastav_popis		integer	10	false	false	false			Skupinastav_popis		varchar	255	false	false	false		
		DIC		integer	10	false	false	false	false		Stav_klienta		varchar	255	false	false	false		
		Ess_ID		integer	10	false	false	true	false		STD_děle		char	1	false	false	false		
		Ess_Nazev		integer	10	false	true	false	false		STD_effdate		date	0	false	false	false		
		Ess_obrat_CZK		integer	10	false	true	false	false		ZDROJYS_ID		integer	10	false	false	false		
		Ess_obrat_datum		date	0	false	true	false	false										
		Ess_obrat_druh		varchar	255	false	true	false	false										
		ICO		varchar	20	false	false	true	false										
		ID_Klienti		integer	10	true	false	true	true										
		KLIENTI_FAKTA_ID		integer	10	false	false	false	false										
		NACE		varchar	10	false	false	false	false										
		NACE_popis		varchar	10	false	true	false	false										
		Nazev_firmy		integer	10	false	false	false	false										
		OBOR_PODNIKARNI_ID		integer	10	false	false	false	false										
		Obrat_firmy_datum		date	0	false	true	false	false										
		Obrat_firmy_CZK		integer	10	false	true	false	false										
		Obsluha_jmeno		varchar	255	false	false	false	false										
		Obsluha_pozice		varchar	255	false	false	false	false										
		Obsluha_pozice_kod		integer	10	false	false	false	false										
		Pobocka_kod		integer	10	false	false	true	false										
		Pobocka_kod		integer	10	false	false	true	false										
		Pobocka_nazev		varchar	255	false	false	false	false										
		Pobocka_nazev		varchar	255	false	true	false	false										
		Pobocka_popis		varchar	255	false	false	false	false										
		Pobocka_region		varchar	255	false	false	false	false										
		Pocet_zaměstnancu		integer	10	false	true	false	false										

Obrázek 23- metadata původní tabulky Klienti



### 1.5.1 Rozdílné pohledy

Běžným business uživatelům, kteří potřebují pracovat s daty pouze a nějakým způsobem je reportovat, bude určitě více vyhovovat přístup k co nejvíce datům na jednom místě. Řešením může být využití nějakého robustního reportovacího nástroje, kde se všechny data zobrazí na jednom místě a uživatelé s nimi poté mohou lépe pracovat. Bohužel se zde nedá mluvit o tom, že bychom u tabulek, jakkoliv dodržovali normalizační formy.

Pokud se však podíváme na use case uživatelů, kteří jsou trochu zdatnější v získávání dat přímo z tabulek pomocí SQL, tak najednou není potřeba mít data na jednom místě v jedné tabulce, ale lze je tedy udržovat v normalizační formě, a to prvního stupně.

Z pohledu datového analytika chceme samozřejmě mít dostupná všechna data, ovšem také v rozumném formátu a čase. V tomto určitě pomůže správný design tabulek a jejich vazeb, proto v tomto přístupu chceme udržet až 3. normalizační formu.

## Závěr

Když budeme vycházet z našeho návrhu a poznatků, které lze získat v praxi při práci s uživateli, tak si můžeme všimnout, že musíme nalézt kompromis. Kompromis, který zajistí řešení pro všechny uživatelské potřeby, protože na jedné straně vycházíme z potřeby dostupnosti dat na jednom místě a na té druhé chceme mít vše roztríděno a v elementární podobě.

Nejlepším řešením je vytvořit rozdílná prostředí pro všechny tyto potřeby, protože poté můžeme využít to nejlepší z každého přístupu. Pokud k naší databázi přistoupíme od základu zodpovědně, uděláme si návrh a zhmotníme ho v ERD, pak můžeme odhalit již na úplném začátku nedokonalosti v návrhu tabulek, nebo jejich vazeb. V dalším kroku pak založíme v databázi tabulky dle normalizačních norem tak, aby i jejich dostupnost byla co největší z pohledu strojového času.

Pro uživatele, kteří dokážou zpracovat výstupy se základními znalostmi SQL, můžeme podklad pro reporty vytvořit v dalším prostředí nebo dokonce softwaru – aktuálním trendem je třeba využití Keboola. Tento software nám lehce umožní propojení na stávající databázové řešení a nezávisle na něm ve svém prostředí lze pracovat s daty a dělat různé transformace, agregace a zároveň je zde zajištěna hladká integrace. Díky tomu se nezatíží samotný reportovací nástroj.

Posledním stádiem bude samotný reportovací nástroj, do kterého budeme načítat již zpracovaná data a tím zajistíme nižší nároky na reportingový nástroj a zároveň budou mít business uživatelé snadnější práci při tvorbě reportů nebo využívání samotných reportů a filtrů.

## 4. Seznam použitých zdrojů

## 5. Bibliografie

- Biscobing, J. (17. 02 2023). *What is Entity Relationship Diagram (ERD)? | Definition from TechTarget*. Načteno z TechTarget:  
<https://www.techtarget.com/searchdatamanagement/definition/entity-relationship-diagram-ERD>
- DataSunrise. (02. Říjen 2022). *What is Partitioning? | DataSunrise - Data & DB Security*. Načteno z What is Partitioning?: <https://www.datasunrise.com/professional-info/what-is-partitioning/>
- Malysheva, V. (18. 02 2023). *What Is Data Reporting And How To Create Data Reports For Your Business | OWOX BI*. Načteno z OWOX BI:  
<https://www.owox.com/blog/articles/data-reporting/>
- Oracle. (02. 10 2022). *Partitioning Overview*. Načteno z Oracle:  
<https://www.oracle.com/database/technologies/partitioning.html>
- Pokorný, J. (1992). *Databázové systémy a jejich použití v informačních systémech* / Jaroslav Pokorný. Praha, Praha, Česká republika.
- Vávrová, I. M. (23. Říjen 2022). *cirkomo*. Načteno z Barva jako nástroj designéra:  
<https://www.cirkumo.cz/barva-jako-nastroj-designera-2/>
- Vostrovský, V. (2014). *Vytváření databází v oracle*. Praha: Česká zemědělská univerzita v Praze Provozně ekonomická fakulta.
- Zendulka, J. (16. 12 2003). *Databázové systémy - 5 Formalizace návrhu databáze*. Brno, Jihomoravský kraj, Česká republika.

## 6. Seznam obrázků, tabulek, grafů a zkratk

### 1.6 Seznam obrázků

Obrázek 1 - schéma datových modelů .....	12
Obrázek 2 - Příklad relační tabulky .....	13
Obrázek 3 - schéma partitioning .....	17
Obrázek 4 - Příklad ERD .....	18
Obrázek 5 - Kroky pro report .....	20
Obrázek 6 - Rady pro tvorbu reportu .....	21
Obrázek 7 - komplementární barvy .....	22
Obrázek 8 - analogické barvy .....	22
Obrázek 9 - split – komplementární .....	22
Obrázek 10 - ERD návrh .....	24
Obrázek 11 - metadata tabulky klienti .....	25
Obrázek 12 - metadata tabulky ESS .....	26
Obrázek 13- metadata tabulky Produkty .....	28
Obrázek 14- metadata tabulky Produkty_mesic .....	30
Obrázek 15- metadata tabulky Pobočka .....	31
Obrázek 16- metadata tabulky Klienti_fakta .....	33
Obrázek 17 - metadata tabulky Adresy .....	34
Obrázek 18 - metadata tabulky Obsluha .....	35
Obrázek 19- metadata tabulky Zdroj_sys .....	36
Obrázek 20 - metadata tabulky Segmentace_klienta .....	37
Obrázek 21 - metadata tabulky Segmentace_fakta .....	38
Obrázek 22 - metadata tabulky Obor_podnikani .....	38
Obrázek 23- metadata původní tabulky Klienti .....	40

### 1.7 Seznam tabulek

Tabulka 1 - příklad 1NF .....	14
Tabulka 2 - příklad 2NF .....	14
Tabulka 3 - příklad 3NF .....	15