Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství

Bakalářská práce

Optimalizace datových základen v relačně databázových evidencích

Jiří Spitzer

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Jiří Spitzer

Informatika

Název práce

Optimalizace datových základen v relačně databázových evidencích

Název anglicky

Optimization of relational database evidences

Cíle práce

Bakalářská práce je zaměřena na problematiku optimalizace databázově koncipovaných informačních zabezpečení. Náplní této této práce je:
a) objasnit teoretické principy relačně databázové technologie v kontextu s problematikou optimalizace databázových evidencí,
b) zmapovat momentální stav této problematiky a vymezit její relevantnost včetně požadavků na ni kladených,
c) navrhnout možnosti přijatelných řešení těchto požadavků,
d) ověřit funkčnost navržených záležitostí,
e) ověřené záležitosti zobecnit pro další možná uplatnění.

Metodika

Použitá metodika zadané bakalářské práce bude založena na studiu a analýze dostupných informačních zdrojů a existujících řešení v dané oblasti. Stěžejními metodami této práce budou metody a techniky relačně databázové technologie a SQL. Navrhované řešení bude zohledňovat identifikované požadavky a očekávání spojená s řešenou záležitostí. Na podkladě syntézy teoretických poznatků a dosažených výsledků budou formulovány závěry této bakalářské práce a následně zobecněny pro další možná použití.

Závazný harmonogram:

Teoretické principy, literární rešerše – předmět 1. zápočtu z BP: do 5.9.2017

Zmapování momentální situace řešené problematiky, identifikace požadavků na ni kladených – do 31.11.2017

Navržení možného řešení se zřetelem na identifikované požadavky – předmět 2. zápočtu z BP: do 10.1.2018

Ověření a zobecnění navrhovaných záležitostí – předmět 3. zápočtu z BP: do 12.3.2018

Doporučený rozsah práce

45-55

Klíčová slova

datová základna, relačně db evidence, datová normalizace, datová integrita, SQL

Doporučené zdroje informací

BEGG, C., CONOLLY, T., HOLOWCZAK, R.: Mistrovství databáze, profesionální průvodce tvorbou efektivních databází. Computer Press. Brno 2009. ISSBN 978-80-251-2328-7

HERMANDEZ, M.: Návrh databází, GRADA 2005. ISBN 80-247-0900-7

HOTEK, M.: Microsoft SQL Server 2008 : krok za krokem. Computer Press. Brno, 2009. ISBN 978-80-251-2466-6

JORGENSEN, A.: Microsoft SQL server 2012 bible. 2012. Wiley. Indianapolis, 2012. ISBN 978-1-118-10687-7

MOLINARO, A.: SQL Kuchařka programátora. Computer Press. Brno 2009. ISBN 978-80-251-2617-2

POKORNÝ, J.: Databázové systémy. ČVUT Praha 2013. ISBN 978-80-01-05212-9

TEOREY, T.J.: Database modeling and design: logical design. 5th ed. Burlington, MA: Morgan Kaufmann, 2011. ISBN 9780123820204

Předběžný termín obhajoby

2017/18 LS – PEF

Vedoucí práce

doc. Dr. Ing. Václav Vostrovský

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 11. 1. 2018

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 11. 1. 2018

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 12. 03. 2018

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Optimalizace datových základen v relačně databázových evidencích" 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 14.3.2018 _____

Poděkování

Rád bych touto cestou poděkoval doc. Ing. Václavu Vostrovskému, Ph.D. za odborné vedení při vypracování této bakalářské práce.

Optimalizace datových základen v relačně databázových evidencích

Abstrakt

Bakalářská práce je tematicky zaměřena na problematiku optimalizace datových základen v relačních databázích. Teoretická východiska se opírají především o metodologii vývoje databázových systémů a princip relačního modelu dat. Stěžejní oblasti zahrnují sběr a analýzu požadavků, metodiku datového modelování na konceptuální i logické úrovni, datovou normalizaci a návrh integritních omezení. Na základě teoretických poznatků a zmapování současného stavu je řešena reálná problémová situace, jejímž výstupem je normalizované schéma relačního modelu dat včetně navržených integritních omezení.

Klíčová slova: datová základna, relační databáze, datová normalizace, datová integrita, datové modelování

Optimization of relational database evidences

Abstract

Bachelor thesis is focused on optimization of relational database structurer. Theoretical part is based primarily on database systems development methodology and principles of relational data model. Core areas are gathering requirements and analysis, methodology of data modeling on conceptual and logical level, database normalization, data integrity. Practical part is focused on database structure optimisation in real company. Solution is based on theoretical scope and analysis of current situation. Result of this part is normalized relational schema including integrity constraints.

Keywords: database, relational database, database normalization, data integrity, data modeling

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Informační tok, respektive informační zpětnovazební smyčka, představuje z pohledu Systémové dynamiky důležitý aspekt, jehož působení výrazně ovlivňuje chování celého systému. Úplná absence informační vazby, případně její nevhodná realizace, bývá častou příčinou nežádoucího chování systému.

Rozvoj v oblasti informačních a komunikačních technologií se výrazně podílí na formování obrazu současné společnosti. V důsledku rozšíření výpočetní techniky a rozvoje globální počítačové sítě je denně generováno značné množství dat, která je třeba zpracovávat. Problematika zpracování a analýzy dat patří především v podnikové sféře k aktuálním a často diskutovaným tématům. V návaznosti na rostoucí objem dat spolu se zvyšujícími se požadavky na rychlost a efektivitu jejich zpracování, investují organizace do vývoje, respektive zavedení, softwarově koncipovaných podpůrných nástrojů, jejichž základní komponentou obvykle bývá databázový systém zajišťující kontinuální víceuživatelský přístup a online zpracování transakcí. Data z takto koncipovaných provozních databází mohou být dále využita pro analytické účely s cílem lepšího poznaní prostředí, ve kterém organizace působí. Extrakce a interpretace informací z velkého objemu dat za účelem rozšíření znalostní báze a zlepšení rozhodovacích procesů je předmětem Business inteligence. Základním nástrojem Business inteligence je datový sklad, který zajišťuje jednotný pohled na podniková data z dostupných zdrojů, bez ohledu na jejich různorodou strukturu.

Vhodně koncipovaný informační systém umožňuje efektivní práci s daty, zajišťuje jejich dostupnost v odpovídající podobě a napomáhá k automatizaci rutinních procesů. V důsledku toho je možné zrychlit a zjednodušit úkony uvnitř organizace, ale také interakci s okolními subjekty. Nejen optimalizace procesů na provozní úrovni, ale také analýza dostupných dat může vyústit ve významnou konkurenční výhodu. Efektivita celého systému je však přímo úměrná především důslednému vymezení jednotlivých procesů a konstrukci databázového systému, jakožto hlavní komponenty. Koncepce databázového systému bývá zejména v případě rozsáhlých a komplexních systémů náročný proces nejen z časového, ale i finančního hlediska. Management, který se zdráhá investovat adekvátní množství časových i finančních prostředků do řádného a metodického procesu vývoje databázového systému obvykle musí v budoucnu vynaložit mnohem více

prostředků na odstranění potíží a nedostatků, které mohou mít negativní dopad na chod celé organizace.

Dle obecné definice se informační systém skládá z hardwarového a programového vybavení, procesů, dat a lidí. Na základě této elementární dekompozice lze formulovat hlavní příčiny nežádoucího chování celého systému. V první řadě se může jednat o nevhodně navržené aplikační programy. Důležitým aspektem není pouze zajištění požadované funkcionality, ale také odpovídající uživatelské rozhraní. Zásadním problémem je pak vysoká latence nebo dočasná nedostupnost dat na provozní i rozhodovací úrovni, která může být způsobena nedostatečnými hardwarovými prostředky, chybnou konstrukcí aplikačního programu, nedostačujícím databázovým strojem nebo špatně konstruovanými SQL dotazy. Zvláště důležité aspekty představuje vymezení business procesů a konstrukce datové základny.

V případě nedostatečného technického vybavení nebo špatně napsaných SQL dotazů je problém ihned patrný a relativně snadno identifikovatelný. Současný trh navíc nabízí množství monitorovacích a ladících nástrojů. V případě chybně navrženého procesu nebo nevhodné koncepce datové základny se mohou negativní dopady viditelně projevit s velkou časovou prodlevou a s mnohem horšími následky. Přesto jsou tyto činnosti v podnikové praxi často podceňovány. Nevhodně koncipovaná datová základna může znamenat ztrátu datové integrity, tedy i informační hodnoty. Závažný problém představuje zejména ztráta konzistence dat v důsledku nežádoucí redundance. Důležitým aspektem je rovněž vymezení integritních omezení, které je rovněž předmětem adekvátního návrhu datové základny. Porušení datové integrity může znamenat ztrátu důležitých údajů. Mezi méně závažné dopady nevhodného návrhu datové základny lze řadit obtížnou práci s datovým modelem včetně jeho aktualizace a konstrukce dotazů, procedur a integritních omezení. S tím souvisí i rychlost jednotlivých dotazů. Soubory databáze mohou navíc v případě nevhodného modelu zabírat neúměrně velké místo na disku.

Z výše uvedených skutečností jednoznačně vyplývá smysl a důležitost často podceňovaného návrhu datové základny, který může významným způsobem ovlivnit fungování celého informačního systému.

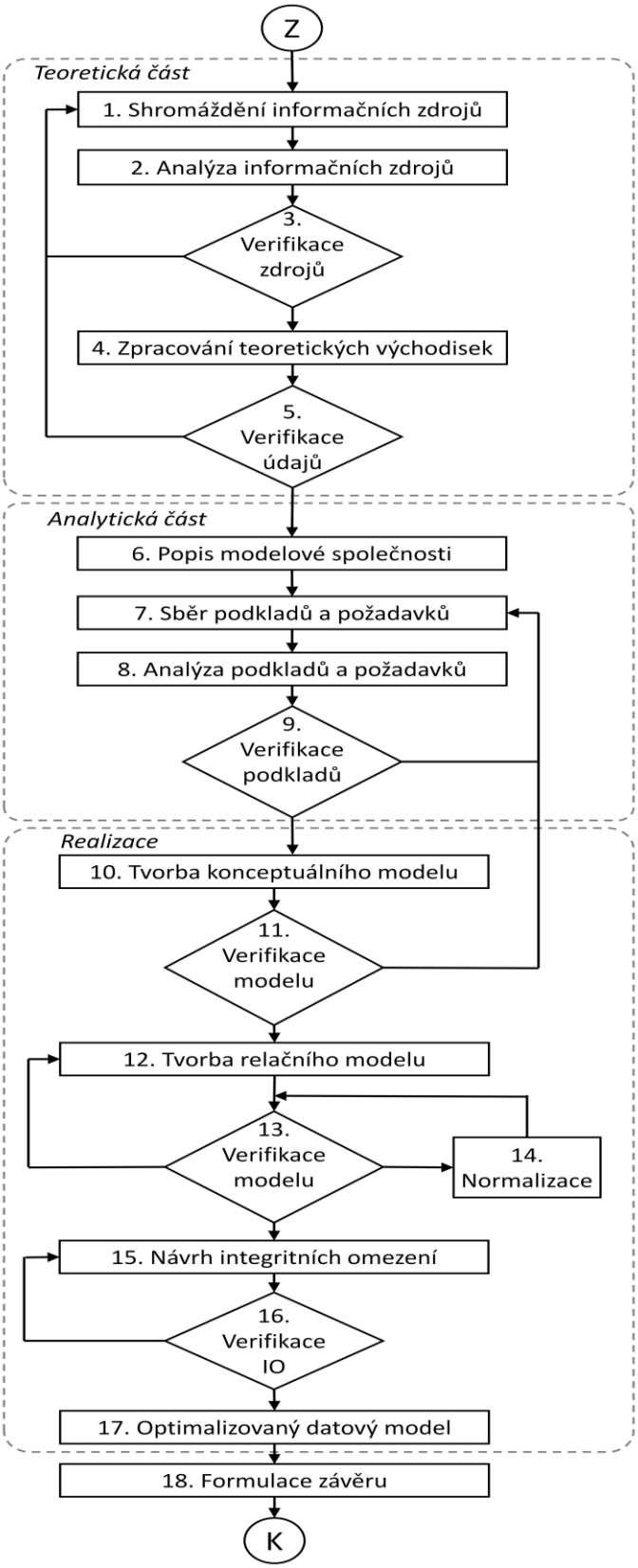
2 Cíl práce a metodika

2.1 Cíl práce

Bakalářská práce je zaměřena na problematiku datového modelování a optimalizaci datových základen založených na relačním modelu dat. Hlavním cílem práce je realizace návrhu adekvátní restrukturalizace nevhodně koncipované datové základny za účelem zabezpečení optimální práce s daty, zajištění datové integrity a zvýšení robustnosti databázového systému. Dílčí cíle zahrnují objasnění teoretických principů datového modelování a relačně databázové technologie v kontextu problematiky optimalizace datových základen, zmapování aktuálního stavu této problematiky a vymezení její relevantnosti s ohledem na kladené požadavky. Na základně takto zpracovaných podkladů lze navrhnout možná řešení. S ohledem na komplexitu celého procesu optimalizace struktury datové základny je práce zaměřena zejména na datové modelování, jehož výstupem je normalizované schéma relačního datového modelu včetně navržených integritních omezení.

2.2 Metodika

Metodika bakalářské práce je založena na studiu a analýze dostupných informačních zdrojů a existujících řešeních v oblastech relačně databázové technologie, vývoje databázových systémů, a datového modelování. Na podkladě získaných informací je vypracován teoretický základ práce včetně vymezení optimálního postupu pro realizaci praktické části. Základem pro řešení reálné problémové situace je sběr a analýza požadavků zadavatele, která je založena především na přímém dotazování odpovídajících osob, analýze současné datové základny a pozorování organizace za provozu. Následuje část konceptuálního modelování. Prověřený konceptuální model je následně transformován do struktury relačního modelu dat. Uvedená schémata jsou realizována prostřednictvím Diagramu tříd v notaci UML 2.0. Následně je testováno splnění třetí normální formy. V posledním kroku jsou vymezena integritní omezení. Na podkladě syntézy teoretických poznatků a dosažených výsledků jsou formulovány závěry bakalářské práce, které jsou následně zobecněny pro další možná použití.



Obrázek 1 – Schématické znázornění zvoleného metodického postupu realizace této bakalářské práce (zdroj: autor)

3 Teoretická východiska

První část práce je zaměřena na stěžejní teoretická východiska, která jsou základem pro samotné řešení problematiky optimalizace datových základen v relačně databázových evidencích.

3.1 Databázový systém

Databázový systém (DBS) je souhrnné označení pro bázi dat, systém řízení báze dat (SŘBD) a soubor aplikačních programů interagujících s databází prostřednictvím SŘBD. [2]

Databázi lze obecně chápat jako kolekci vzájemně souvisejících dat sloužící informačním potřebám. Organizované uložení dat v relační databázi umožňuje snadný přístup k těmto datům, jejich správu a aktualizaci. Databáze tedy slouží organizacím k uchovávání integrovaných, sdílených a takzvaně perzistentních dat (životnost dat přesahuje běh aplikace), jejichž zpracováním lze získat hodnotné informace. [1]

Systém řízení báze dat představuje softwarové rozhraní mezi aplikačními programy a samotnou bází dat. Řídí přístup k datům, umožňuje vytvářet, spravovat a aktualizovat databázi, zajišťuje ochranu před neoprávněným přístupem, kontrolu konzistence uložených dat a poskytuje řadu dalších užitečných nástrojů v závislosti na konkrétním dodavateli. K datům lze přistupovat pouze prostřednictvím tohoto aparátu. Vzhledem k centralizovanému uložení dat musí SŘBD umožňovat současný přístup více uživatelů. Komunikačním prostředkem je zpravidla dotazovací jazyk. [1]

3.2 Relační databáze

Koncept relačního datového modelu vznikl na přelomu 60. a 70. let 20. století v reakci na značné nedostatky tehdy využívaných souborově orientovaných systémů pro hromadné zpracování dat. Model, založený na teorii množin a predikátové logice prvního řádu, poprvé formálně představil doktor E. F. Codd v roce 1970 ve své práci „A relational model of data for large shared data banks“. K velkému rozšíření relačních databází dochází v 80. letech 20. století, kdy na trh vstoupila společnost Oracle se stejnojmenným produktem. [2][7]

Databázový přístup se vyznačuje zejména:

a) vysokým stupněm nezávislosti dat
b) odstraněním nežádoucí redundance
c) nástroji pro kontrolu konzistence
d) sdílením centrálně uložených dat
e) zabezpečeným přístupem k datům
f) abstrakcí dat

V důsledku oddělení logické struktury databáze od fyzického uložení dat je logický návrh nezávislý na fyzické implementaci a zároveň lze provádět změny v logické struktuře bez nutnosti přepsání aplikačních programů. Tato koncepce umožňuje snadné rozšíření struktury databáze bez dopadu na uchovávaná data. Centralizované uložení sdílených dat umožňuje minimalizaci nároků na kapacitu paměťových médií a snazší aktualizaci dat. Především však vede k omezení redundantních dat, čímž výrazně přispívá k zajištění konzistentních údajů. Aby mohla být data sdílena napříč organizací je nezbytné zajistit souběžný přístup více uživatelů s různými úrovněmi oprávnění. Za tímto účelem je možné přistupovat k datům pouze prostřednictvím SŘBD a nikoli přímo. SŘBD dále zajišťuje centrální víceúrovňovou kontrolu integrity dat. Předností relačně-databázového přístupu je také jednoduchá reprezentace dat v podobě dvourozměrných tabulek, se kterými dokáže uživatel snadno pracovat. Uživatel pracuje pouze s datovými strukturami na vyšší úrovni abstrakce a nemusí znát fyzické uložení dat na disku. [2][7]

3.3 Životní cyklus vývoje databázového systému

Životní cyklus vývoje databázového systému představuje ověřenou posloupnost činností včetně definice aktivit, technik a nástrojů, kterou je vhodné při vývoji dodržet. Metody vývoje jsou založeny na poznatcích a zkušenostech designerů a mohou se od sebe formálně lišit. Obecný postup je však většinou ve své podstatě stejný a lze ho využít nejen při vývoji nového databázového systému, ale také při optimalizaci stávajícího. [1][7]

Níže uvedený obecný postup je formulován s ohledem na cíle bakalářské práce.

3.3.1 Plánování a definice systému

Fáze plánování zahrnuje úkony managementu cílené na zajištění efektivního průběhu jednotlivých fází vývoje databázového systému. V rámci této fáze by mělo dojít k formulaci celkových cílů a účelu databázového systému, vymezení dílčích cílů, odhadu celkové práce, zdrojů a finančních prostředků. [1][7]

V dalším kroku je třeba vymezit rozsah a hranice zkoumaného systému, identifikovat a definovat možné interakce s okolními subsystémy a definovat uživatelské pohledy, které vymezují, co od databázového systému požadují konkrétní pracovníci či jednotlivé útvary organizace. [1][7]

3.3.2 Sběr a analýza požadavků

Sběr a analýza požadavků je klíčovou, ale často podceňovanou, součástí návrhu databázového systému. Bez odpovídající znalosti dané problematiky není možné navrhnout požadovanou strukturu databáze, která by řádně plnila informační potřeby organizace. Při osvojování problematiky hraje důležitou roli zajištění komunikační vazby se všemi zainteresovanými skupinami a shromáždění relevantních podkladů pro analýzu. [1][7]

U zavedených organizací, které již určitým způsobem s daty pracují lze k identifikaci informačních potřeb využít poznatky o charakteru a účelu uchovávaných dat. Struktura nové databáze by však neměla vycházet ze struktury stávající databáze. [1]

Pro sběr informací neexistuje jediná správná cesta. Existuje mnoho efektivních metod, jejichž použití závisí vždy na konkrétních parametrech projektu a je ovlivněno řadou faktorů. Především u větších projektů je však žádoucí využít ke sběru informací více prostředků. Všechny specifikované požadavky je třeba řádně zdokumentovat a nechat odsouhlasit managementem organizace, nejlépe standardizovanou cestou. [1][2]

K nejčastějším formám získávání podkladů patří rozhovory, jejichž výhodou je přímá interakce se zainteresovanými osobami umožňující dotazovaným osobám volně odpovídat na kladené otázky a rozvíjet myšlenky. Tazatel může pozorovat neverbální komunikaci, reagovat na odpovědi a klást doplňující otázky. Je zde prostor pro vyjasnění případných nesrovnalostí z obou stran. Vedení rozhovoru je však časově náročné a vyžaduje dobré komunikační schopnosti tazatele, který musí být připraven na protichůdné názory a zájmy zapojených osob nebo na jejich neochotu spolupracovat. [1][2]

V případě potřeby zajištění podkladů od velkého počtu respondentů je vhodné provést dotazníkové šetření. Shromažďování informací prostřednictvím dotazníkových formulářů umožňuje dotazované osobě formulovat odpovědi v klidu, soukromí a často také anonymně. Respondenti tak mohou vyjádřit názory, které by se v případě přímého dotazování zdráhali říci. Dotazníky s pevným formátem (tj. výběr ze specifických odpovědí) lze hromadně zpracovávat prostřednictvím softwarových nástrojů, které umožňují snadnou a rychlou analýzu. Nevýhodou absence přímého kontaktu mezi analytikem a respondenty je nemožnost reakce na případné neporozumění kladené otázce na straně jedné i uvedené odpovědi na straně druhé. Struktura formuláře je velice důležitá a její příprava může být časově náročná. Analytik zároveň nemá možnost motivovat účastníky k aktivnímu zapojení, ani sledovat jejich neverbální komunikaci. [1][2]

Požadavky kladené na systém lze získávat rovněž na základě pozorování organizace za provozu. Analýza rutinních činností často poskytne důležité informace o systému. Analytik se může kromě pozorování koncových uživatelů sám podílet na běžných činnostech, ke kterým je systém využíván a které by měl podporovat. [1][2]

3.3.3 Návrh databáze

Metodologie návrhu databáze představuje standardizovaný a strukturovaný přístup obsahující soubor procedur, technik a nástrojů pro podporu a usnadnění procesu návrhu, který lze rozčlenit do tří hlavních fází:

1) Konceptuální návrh
2) Logický návrh
3) Fyzický návrh

Během fáze konceptuálního návrhu jsou sjednoceny všechny vnější uživatelské pohledy a požadavky na systém kladené. Na základně analýzy získaných informací je následně modelována obecná struktura databáze, nezávislá na fyzickém uložení dat. Výsledkem procesu je formální popis modelované reality, který reflektuje požadavky organizace jako celku. Takto formalizované vyjádření logické struktury uložení dat se nazývá konceptuální datový model. [3][4]

Během fáze logického návrhu je obecný konceptuální model transformován do podoby logické reprezentace databáze. Touto podobou může být například relační datový model, který je reprezentován množinou relačních tabulek. Každá tabulka následně projde normalizačním procesem a na závěr jsou definována integritní omezení.

Během fáze fyzického návrhu je definováno, jakým způsobem bude logický návrh fyzicky implementován v prostředí konkrétního SŘBD. V této fázi jsou definovány podkladové tabulky, organizace souborů na disku, indexy, uživatelské pohledy a bezpečnostní mechanismy. [3]

3.3.4 Návrh aplikací

Návrh aplikačních programů je realizován souběžně s návrhem databáze. Tyto procesy jsou na sebe závislé a je třeba mezi nimi zajistit komunikační vazbu. [1][7]

Cílem návrhu aplikací je zajištění požadované funkčnosti a design přívětivého, intuitivního a snadno ovladatelného uživatelského rozhraní. Návrh aplikace může být podpořen tvorbou prototypu. Jedná se o zjednodušený model navrhovaného systému, který obvykle nedisponuje všemi vlastnostmi a funkcemi, ale jeho tvorba je zpravidla časově i finančně nenáročná.

Za účelem zajištění interakce mezi aplikačním programem a databázovým systémem je třeba definovat návrh transakcí. Transakce zde představuje operaci provedenou uživatelem (aplikačním programem), během které dochází k vyvolání nebo aktualizaci uložených dat, eventuálně k obojímu. [1]

3.3.5 Testování a implementace

Před nasazením databáze do ostrého provozu je vhodné provést důkladné testování za účelem odhalení maximálního množství možných chyb. Zejména u komplexních systémů je žádoucí provádět proces testování systematicky a metodicky. Databáze by měla být testována na realistických datech, nejlépe se zapojením koncových uživatelů. [1][7]

Implementace představuje fyzickou realizaci navržených struktur databáze prostřednictvím jazyka pro definici dat (případě grafického nástroje) v konkrétním SŘBD, realizaci aplikačních programů, zabezpečení databáze a dalších náležitostí. [3]

V případě, že organizace již nějakým způsobem data uchovává je třeba zajistit načtení z aktuálních datových zdrojů do nové databáze. Původní data ale mohou být uložena v jiných strukturách a formátech. V takovém případě je nutné provést jejich konverzi. V současné době SŘBD obvykle disponují nástrojem, který umožňuje snadné načtení dat včetně jejich konverze.

3.3.6 Provozní údržba

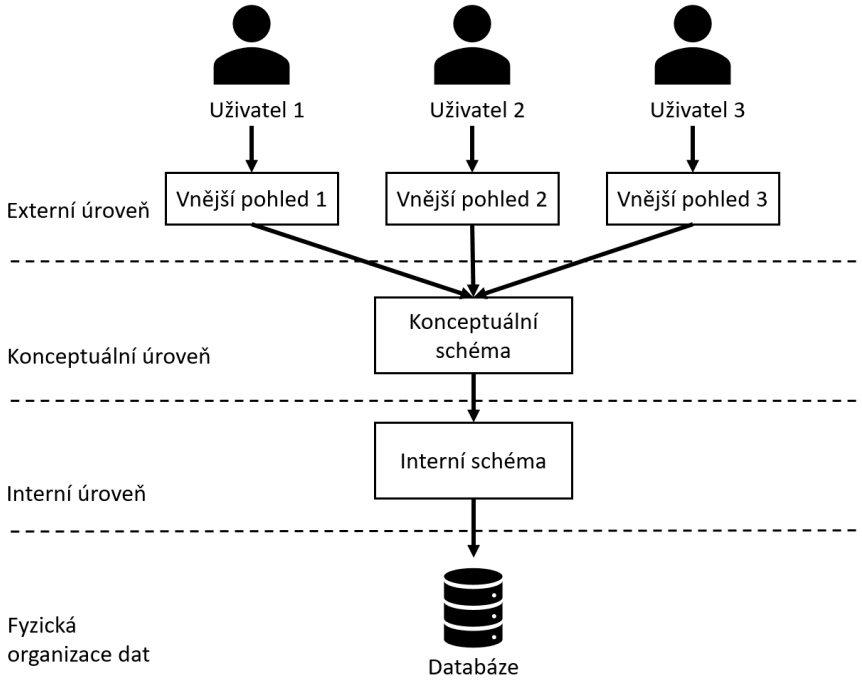
Po nasazení systému do ostrého provozu začíná fáze monitorování a údržby. Monitorovat lze například propustnost transakcí, dobu odezvy či kolik místa na disku je potřeba pro uložení databázových souborů. Ukáže-li se během provozu návrh databáze jako nevhodný nebo dojde-li ke změně původních požadavků na ni kladených, je vhodné provést optimalizaci. Přínosem může být i aktualizace SŘBD či jiných komponent. [3][7]

3.4 Třívrstvá architektura ANSI-SPARC

Podle třívrstvé architektury ANSI-SPARC, navržené v polovině sedmdesátých let, lze popsat datové položky na třech různých úrovních abstrakce:

1) Externí úroveň
2) Konceptuální úroveň
3) Interní úroveň

Externí úroveň představuje řadu vnějších uživatelských pohledů na práci s daty, které se liší v závislosti na potřebách jednotlivých uživatelů. Konceptuální úroveň představuje ucelený pohled na logickou strukturu databáze, nezávisle na fyzickém uložení dat. Interní úroveň pak popisuje fyzickou implementaci databáze. [1]



Obrázek 2 – Třívrstvá architektura ANSI-SPARC (zdroj: autor)

Cílem třívrstvé architektury je umožnění logické a fyzické nezávislosti dat. V případě logické nezávislosti dat hovoříme o možnosti změny logického návrhu bez nutnosti přepisovat aplikační programy. Fyzická nezávislost představuje odolnost konceptuálního schématu v případě přechodu databáze na jinou platformu. [1]

3.5 Konceptuální návrh

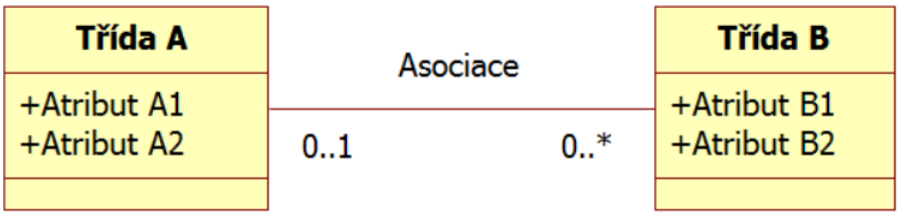
Konstrukce datového modelu představuje mentálně náročnou činnost především v případě větších projektů, kde je zapotřebí pracovat s velkým množstvím objektů, mezi kterými existují komplexní vztahy. Za účelem zajištění odpovídající kvality modelu je vhodné postupovat při modelování systematicky. Obecně lze vycházet ze dvou základních přístupů:

a) Zdola nahoru
b) Shora dolů [1][3][4]

V případě strategie „zdola nahoru“ dochází nejprve vymezení všech atributů, které jsou v dalších fázích seskupovány do vyšších celků. Z důvodu přehlednosti lze tento postup doporučit pouze pro modely s nanejvýš 25 atributy, což je v praxi většinou nedostatečné.

U komplexních projektů se v praxi osvědčil postup „Shora dolů“. V případě této strategie je naopak nejprve nastíněna hrubá struktura modelu, která je následně zjemňována až do finální podoby.

Při návrhu databáze se ke znázornění dat na konceptuální úrovni používá Entity-relationship diagram, jehož základy zformuloval Peter Chen v roce 1976. Za dobu své existence prošel model mnohostranným vývojem a vznikla celá řada platných konvencí zápisu. Z tohoto důvodu je výstižnější hovořit o celé rodině ER modelů. V současné době se však stává preferovanou volbou mladší standardizovaný modelovací nástroj UML. [3][4][5]



Obrázek 3 - Notace UML (zdroj: autor)

3.5.1 Standard UML

Unified modeling language (UML) představuje standardizovaný grafický modelovací jazyk vyvinutý v reakci na speciální potřeby objektově orientovaného přístupu. Cílem je zjednodušení komunikace během fází návrhu a vývoje softwaru. V současné době je UML rozšířeným standardem používaným pro vizualizaci, specifikaci, návrh a dokumentaci aplikačních programů a softwarových systémů. [4][5][8]

UML 2.0 definuje třináct typů diagramů, které lze rozčlenit do tří kategorií:

a) Strukturální diagramy
b) Diagramy chování
c) Diagramy interakce

K základním a nejčastěji využívaným UML diagramům patří Class diagram (diagram tříd), který je využíván k modelování statické struktury objektů, o nichž chce organizace uchovávat informace, jejich vlastností a vzájemných vztahů. Z důvodu své jednoduchosti a srozumitelnosti je velice dobře uplatnitelný během fáze sběru požadavků a analýzy ke komunikaci se zadavatelem a koncovými uživateli. Pro tento koncept jsou charakteristické tři základní prvky: třída, asociace a atribut. Diagram tříd vychází z Chenova ER modelu a představuje alternativní nástroj k tvorbě konceptuálního modelu dat. [4][5][8]

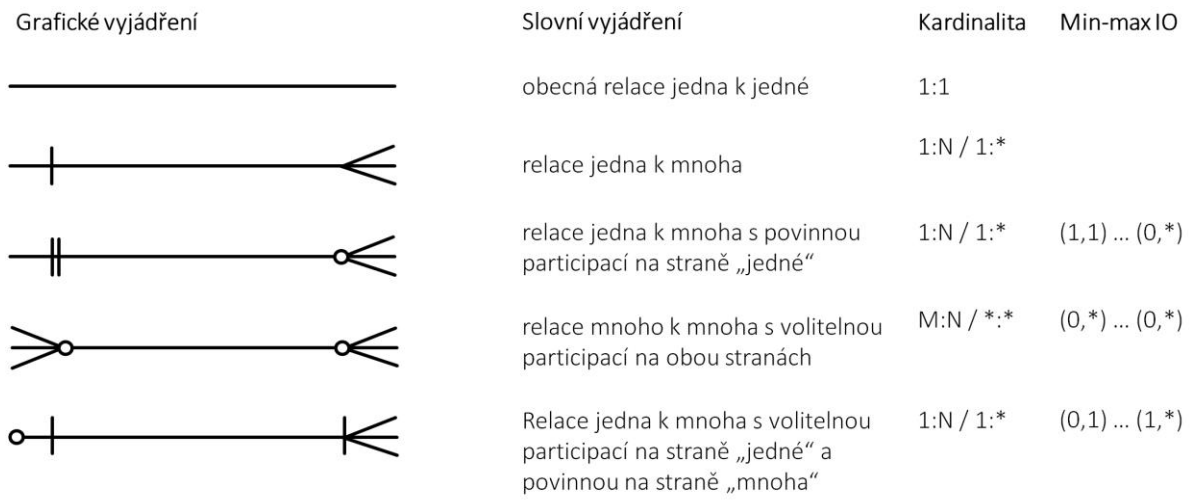
Třída reprezentuje množinu nezávislých a jednoznačně identifikovatelných objektů reálného světa se shodnými vlastnostmi. Může se jednat o objekty fyzické i abstraktní. Každá třída je vymezena jednoznačným pojmenováním, skupinou atributů a skupinou relevantních operací, které lze nad objektem provádět. Možnost specifikace seznamu operací se však během konceptuálního návrhu databáze nevyužívá. Graficky je třída reprezentována obdélníkem rozdělený na tři části. První část je vyhrazena pro jednoznačné pojmenování třídy, v prostřední části se nachází seznam atributů a třetí část je určena pro výčet možných operací. Jednoznačně identifikovatelný výskyt třídy je označován pojmem objekt. V terminologii ER modelů je pro obdobnou konstrukci užíván termín entita. [3][4][5]

Atributy představují charakteristické vlastnosti tříd vymezené určitou množinou hodnot – doménou. Hodnoty atributů mají popisný, respektive identifikační, charakter. Množina atributů jednoznačně identifikující každý výskyt objektu je označována jako superklíč. Obsahuje-li tato množina pouze minimální počet atributů nezbytných k

identifikaci všech objektů, jedná se o klíč kandidátní. Z množiny kandidátních klíčů designer vybere právě jeden primární klíč, ostatní klíče jsou označovány jako alternativní. Na úrovni konceptuálního modelu lze atributy členit na jednoduché a složené. Hodnoty jednoduchých atributů jsou atomické – dále nedělitelné. V případě složených atributů lze hodnoty ještě dekomponovat na menší samostatné části. Další možnou klasifikaci lze provést na základně množství možných výskytů daného atributy, kdy jsou rozlišovány atributy s jednou hodnotou a vícehodnotové. Na rozdíl od ER diagramů je v Class diagramu nutné vyčlenit všechny vícehodnotové atributy do samostatných tříd. To je praktické z hlediska následující fáze modelování, jejímž cílem je realizace logické struktury databáze, kde je vyčlenění vícehodnotových atributů nezbytné. [3][4][5]

Zvláštní pozornost je třeba věnovat vztahům mezi modelovanými objekty. V kontextu ER modelování jsou vztahy označovány pojmem relace. Počet entit participujících na relaci udává stupeň relace. Specifickým případem je rekurzivní relace, kdy se v relaci vyskytuje jedna entita ve více rolích. U každé relace lze určit kardinalitu, tedy možný počet výskytů entit participujících na dané relaci. Lze rozlišit tři základní typy vazeb:

a) One-to-one (Jedna k jedné)
b) One-to-many (Jedna k mnoha)
c) Many-to-many (Mnoho k mnoha) [3][4][5]



Obrázek 4 – Demonstrace principu grafického znázornění vazeb v ER diagramu (zdroj: autor)

Kromě kardinality lze u vztahu určit ještě povinnost participace. Kardinalita a participace vymezují možný rozsah hodnot, kde omezení participace představuje minimální počet výskytů entity v relaci a kardinalita udává počet maximální. Vymezení rozsahu hodnot se nazývá multiplicita relace a představuje důležité integritní omezení, které bývá označované jako min-max IO. Nepovinné členství je vyjádřeno nulou v minimu, povinné jedničkou. [1][4]

Class diagram umožňuje realizaci více typů vztahů mezi třídami. Obecná vazba je označována pojmem asociace a je realizována plnou čárou mezi dvěma třídami. Asociace je zpravidla vhodně pojmenována a může obsahovat seznam atributů, které jsou graficky znázorněny v obdélníku připojenému přerušovanou čárou k čáře spoje. Na obou koncích spojení je možné vyznačit multiplicitu, tedy kardinalitu a omezení participace. Specifičtějším typem vztahu je agregace, pomocí které lze vyjádřit závislost jednoho objektu na druhém. Závislý objekt však může existovat i jako samostatný celek. Přísnější formou závislosti jednoho objektu na druhém je třetí typ vazby označovaný jako kompozice. V případě kompozice nemůže závislý objekt existovat samostatně. Agregace je graficky znázorněna jako prázdný kosočtverec na straně nadřazené třídy, kompozice je znázorněna jako plný kosočtverec. V případě UML mohou být vztahy buď jednosměrné nebo obousměrné, přičemž jednosměrné se v diagramu značí šipkou. [4][5][6]

UML umožňuje modelovat hierarchii. V případě, že model obsahuje množinu tříd s částečně shodnými vlastnostmi, může být vhodné vyčlenit některé společné atributy do specializované třídy na vyšší úrovni hierarchie. Třídy na nižší úrovni pak mohou od této třídy dědit předem vymezené vlastnosti. [3][5]

S procesem dědění atributů se pojí termíny generalizace a specializace. Jedná se o inverzní procesy definice podtříd a nadtříd, přičemž specializace vychází z přístupu „Shora dolů“, zatímco generalizace vychází z přístupu „Zdola nahoru“. Během procesu specializace jsou identifikovány vlastnosti tříd, které je vhodné vyčlenit do množiny podtříd. V případě generalizace jsou naopak identifikovány společné vlastnosti, které je vhodné seskupit do jedné nadtřídy. [3][5][10]

3.6 Logický návrh

Logický návrh představuje druhou fázi návrhu databáze, během které je konceptuální datový model transformován do odpovídající struktury logického návrhu. V případě metodologie formulované v této práci je výstupem logického návrhu normalizované schéma založené na relačním modelu dat, včetně vymezených integritních omezení. Cílem je zajištění odpovídajícího podkladového materiálu pro návrh fyzické implementace databáze. [1][2]

3.6.1 Relační datový model

V relačním datovém modelu jsou data uchovávána v takzvaných relacích (tabulkách), které lze chápat jako dvourozměrné struktury tvořené záznamy a atributy. Každá relace je jednoznačně identifikována svým jménem. Atributy jsou jednoznačně pojmenované sloupce tabulek s pevně vymezenou množinou přípustných hodnot. Pořadí atributů lze vzájemně zaměňovat, je nevýznamné. Množina přípustných hodnot atributu se nazývá doména. U každé domény je nutné určit datový typ, případně lze definovat seznam logických pravidel, označovaný jako integritní omezení. Na základě definice domény může SŘBD zamezit některým chybným transakcím. Záznam (n-tice) je skupina souvisejících polí na které lze pohlížet jako na celek. V relační tabulce musejí být hodnoty v polích atomické (dále nedělitelné). Každý záznam musí být jednoznačně identifikován. V relaci se tedy nemohou vyskytovat duplicitní záznamy. Pořadí záznamů je rovněž nevýznamné. [1][2]

V relační databázi lze budovat vztahy mezi tabulkami buď na základě takzvaných cizích klíčů nebo pomocí asociativních (vazebních) tabulek v závislosti na typu vztahu. Cizí klíč je atribut či skupina atributů jedné relace, které se odkazují na primární klíč jiné relace. [1][2]

Pro práci s daty jsou nezbytné tři základní operace relační algebry: projekce, selekce a spojení. Projekce slouží pro výběr podmnožiny požadovaných atributů z dané relace. Selekce slouží pro vytvoření nové relace obsahující výběr specifických záznamů z původní relace. Operace spojení sjednocuje dvě relace na základě zadaného klíče. [1][2]

3.6.2 Normalizace

Normalizaci lze chápat jako proces vedoucí k zajištění efektivní organizace dat v relační databázi s cílem zamezení nežádoucí redundance dat, minimalizace nároků na paměťové médium a usnadnění nejen případných budoucích změn ve struktuře báze dat, ale také usnadnění vkládání, mazání a výběru dat z databáze. Nejedná se o deterministický proces, může existovat více vyhovujících řešení. [1][9]

Během procesu normalizace je u jednotlivých tabulek relačního datového modelu prověřováno, zda splňují kritéria takzvané normální formy (NF). Celkem existuje šest normálních forem, které si lze představit jako pravidla pro zefektivnění organizace dat v tabulkách relačního datového modelu. Každá normální forma v sobě vždy obsahuje formu nižší. Splňuje-li například relace třetí normální formu, lze tvrdit, že splňuje i první a druhou normální formu. Čím vyšší normální forma, tím lepší by měla být manipulace s daty v databázi. Nesplňuje-li relace kritéria příslušné normální formy, je třeba ji dekomponovat na více relací. [1][9]

Některé normální formy se vztahují pouze na relace se složeným primárním klíčem. Neobsahuje-li relace složený primární klíč, dochází ke splývání normálních forem. Pro relaci, jejíž primární klíč je tvořen pouze jedním atributem tedy platí, že v případě její normalizace do 1NF zároveň automaticky splňuje i 2NF a v případě normalizace do 3NF zároveň automaticky splňuje všechny zbylé normální formy, tedy: BCNF, 4NF a 5NF. Normalizace relací do čtvrté a páté normální formy může navíc v praxi znamenat výrazné snížení výkonu databázového systému. Z těchto důvodů jsou tabulky ve většině případech normalizovány nanejvýš do 3NF, případně do BCNF. [1][9]

První normální forma (1NF) říká, že všechny hodnoty v tabulce musejí být atomické, tedy dále nedělitelné. Elementárnost polí představuje nezbytný předpoklad relačního datového modelu. [9]

Relace se nachází ve druhé normální formě (2NF), splňuje-li kritéria 1NF a zároveň jsou všechny neklíčové atributy závislé na celém primárním klíči. Z toho vyplývá, že 2NF je třeba se zabývat pouze v případě, obsahuje-li relace složený primární klíč. [9]

Relace je ve třetí normální formě (3NF), pokud splňuje kritéria 2NF (a tedy i v 1NF) a zároveň neexistuje závislost mezi dvěma neklíčovými atributy. Jinými slovy to znamená, že žádný neklíčový atribut není tranzitivně závislý na primárním klíči. [9]

Boyce–Coddova normální forma (BCNF) pouze zpřísňuje pravidlo 3NF, proto je někdy také označována jako 3.5NF. Dle definice se relace nachází v BCNF právě tehdy, když splňuje podmínky 3NF a zároveň jsou všechny atributy složeného primárního klíče vzájemně nezávislé.

Předmětem čtvrté normální formy (4NF) je definice závislostí uvnitř složeného primárního klíče. Relace je ve 4NF, pokud je v BCNF a zároveň jsou všechny atributy složeného primárního klíče na sobě nezávislé. [9]

Pátou normální formou (5NF) je třeba se zabývat pouze v případě, že relace obsahuje složený primární klíč o alespoň třech složkách. Relace je v 5NF, pokud je v 4NF a hodnoty primárního klíče neobsahují párové cyklické závislosti. [9]

3.6.3 Integrita dat

Zajištění odpovídající kvality a přesnosti informací, které mohou sloužit například jako podklady pro důležitá manažerská rozhodování, patří ke klíčovým cílům každé organizace. Aby mohla databáze poskytovat svým uživatelům přesné a relevantní informace, musí obsahovat platná, přesná a konzistentní data. Za tímto účelem je třeba během fází návrhu a realizace databázového systému definovat pravidla pro aktualizaci dat. SŘBD následně kontroluje, zda jsou prováděné transakce v souladu s těmito pravidly, která jsou označována jako integritní omezení (IO). Konzistence a přesnost uložených dat v závislosti na takto definovaná pravidla se označuje pojmem integrita databáze. [1][2]

Tři základní typy integrity:

a) Entitní integrita
b) Referenční integrita
c) Doménová integrita

Každý záznam musí být v rámci relační tabulky jednoznačně identifikován odpovídajícím primárním klíčem. Entitní integrita zajišťuje, že se v žádné z komponent primárního klíče na úrovni tabulky nevyskytují neznámé hodnoty NULL nebo hodnoty, které by byly vzhledem k již uloženým záznamům v dané relaci duplicitní. Tímto způsobem je zajištěno, že se v tabulce nebudou vyskytovat duplicitní záznamy. [1][2]

Vztahy mezi logicky souvisejícími relačními tabulkami jsou realizovány prostřednictvím cizího klíče, který se odkazuje na hodnotu klíče rodičovské relace. V

rámci referenční integrity je zajišťována korektnost vztahů na základě vymezení oboru hodnot cizího klíče pouze na množinu hodnot odpovídajícího klíče v rodičovské tabulce a neznámou hodnotu NULL. Aby mohla být zajištěna referenční integrita, je rovněž nezbytné kontrolovat aktualizace a mazání záznamů v rodičovské tabulce. K porušení integrity by došlo v případě aktualizace hodnoty, na kterou se odkazuje cizí klíč podřízené tabulky nebo v případě smazání záznamu obsahujícího takovouto hodnotu. Udržení referenční integrity je možné několika způsoby:

a) Restriktivní omezení
b) Kaskáda
c) Nastavení hodnoty NULL
d) Nastavení implicitní hodnoty

V případě restriktivního omezení je zamezeno aktualizaci hodnoty, respektive smazání záznamu s takovou hodnotou, ke které se pojí záznam v podřízené tabulce. [1]

Kaskáda představuje promítnutí změny v rodičovské tabulce do všech podřízených tabulek. Pokud dojde k vymazání záznamu v rodičovská tabulce, budou automaticky vymazány i všechny záznamy logicky podřízené. V případě změny klíče v rodičovské tabulce se tato změna opět kaskádovitě promítne do všech logicky podřízených záznamů.
V případě transakcí, jejichž výsledkem by byla neplatná hodnota odkazu na rodičovskou tabulku je možnou alternativou k zachování referenční integrity nastavení hodnoty cizího klíče na neznámou hodnotu NULL, případně na implicitní hodnotu odkazující na jiný záznam rodičovské tabulky. [1]

Integrita na úrovni jednotlivých polí je dána doménou, která představuje obor přípustných hodnot pro daný atribut. Na této úrovni lze například vymezit povinná pole, která nesmějí obsahovat hodnotu NULL či vymezit rozsahy přípustných číselných hodnot. Soubor pravidel takto vymezující konkrétní obor hodnot příslušných atributů je označován pojmem doménová integrita. Je zřejmé, že doménová integrita nemůže sama o sobě garantovat správnost údajů, neboť zadávání vstupních dat je zatíženo rizikem lidské chyby. Vhodně navržená integritní omezení však výrazně pomáhají riziko lidské chyby minimalizovat. [1]

3.7 Proces návrhu databáze

Na základě klíčových pojmů z oblasti datového modelování, normalizace a datové integrity, které byly předmětem předešlých kapitol, lze nyní popsat možný průběh návrhu relační databáze. Níže popsaný proces lze vnímat jako obecný rámec založený na bohatých poznatcích z praxe zkušených návrhářů.

1.krok: Identifikace entit

Na základě analýzy specifikovaných uživatelských požadavků jsou identifikovány základní objekty modelované reality, které jsou následně zaneseny jako vhodně pojmenované entity do konceptuálního datového modelu. [1][2]

2.krok: Identifikace vztahů mezi objekty

V druhém kroku je pozornost věnována identifikaci vztahů mezi jednotlivými entitami. Každý vztah by měl být vhodně pojmenován, následně je třeba určit kardinalitu a omezení participace. Je nutné prověřit, zda se v modelu nevyskytují redundantní vztahy, problém zdvojeného přiřazení a problém rozdělených dat. Problém zdvojeného přiřazení vzniká v případě, kdy nepřímá cesta mezi dvěma entitami obsahuje dvě relace typu „jedna k více“. Výskyty entity pak lze propojit pouze pomocí víceznačné cesty. Problém rozdělených dat může nastat, pokud nepřímé spojení mezi dvěma entitami obsahuje nepovinnou participaci. Důsledkem toho může být absence spojení mezi souvisejícími entitami. V rámci tohoto kroku je třeba se zaměřit také na relace s kardinalitou „jedna k jedné“. V případě, že je participace obou entit povinná, je třeba tyto entity sloučit do jedné. [1][2]

3.krok: Identifikace atributů a jejich přiřazení k entitám nebo relacím

Během třetího kroku je zkoumáno, jaké informace o entitách mají být uchovávány. Tyto informace jsou následně reprezentovány v datovém modelu pomocí atributů, které mohou náležet jak entitám, tak relacím. Následně je třeba u jednotlivých atributů určit domény. Během kroku přiřazování atributů může vyplynout potřeba restrukturalizace některé části modelu. [1][2]

4.krok: Určení primárních klíčů

U entit jsou následně identifikovány množiny atributů, jednoznačně identifikující každý výskyt entity – kandidátní klíče. Z této množiny je vybrán vhodný primární klíč, který by měl být složen z minimálního počtu atributů. [2]

5.krok: Specializace / generalizace (volitelný)

V některých případech může být žádoucí pokračovat procesem specializace, respektive generalizace, a rozložit odpovídající entity na vhodné podtypy a nadtypy. Tím však zároveň narůstá složitost celého modelu. [10]

6.krok: Transformace konceptuálního modelu do logické struktury

Hotový konceptuální model je následně transformován do logické struktury. Během procesu transformace jsou nejprve na základě entit tvořeny předběžné relace. Na rozdíl od entit mohou tabulky v relačním modelu obsahovat pouze jednoduché atributy. Složené atributy je tedy nutné dekomponovat na samostatné celky a atributy s více hodnotami je třeba vyčlenit do samostatné tabulky. Následně jsou do nově vzniklé struktury doplněny vztahy. V případě binární relace „jedna k více“ je vztah tvořen přidáním cizího klíče k záznamům v dceřiné tabulce. Tento klíč zde slouží jako odkaz na primární klíč rodičovské tabulky. K realizaci binární vazby „více k více“ a komplexních vztahů mezi více než dvěma tabulkami, je třeba použít takzvanou asociační tabulku, která je tvořena cizími klíči odkazujícími se na všechny participanty dané relace. V případě relací „jedna k jedné“ je třeba k identifikaci rodičovské a dceřiné relace využít omezení participace. Pokud je participace povinná na obou stranách, tabulky je třeba sloučit do jedné. U vztahů, kde je participace povinná pouze na jedné straně binární relace, bude cizí klíč přidán do tabulky s povinnou participací. Poslední variantou je nepovinná participace na obou stranách, kde je určení rodičovské relace volitelné. U rekurzivních relací s kardinalitou „jedna k jedné“ je postup stejný s tím, že v případě nepovinné participace na obou stranách je třeba vytvořit druhou tabulku. [1][2]

7.krok: Normalizace

Struktura navrženého modelu by měla být prověřena na základě pravidel normalizace. Je žádoucí, aby všechny relace odpovídali třetí normální formě. V opačném případě je vhodné provést restrukturalizaci modelu do odpovídající podoby. [3]

8.krok: Integritní omezení

Důležitý krok procesu realizace databázového systému představuje vymezení přesných pravidel pro manipulaci s daty – integritních omezení. Cílem těchto pravidel je minimalizace rizika výskytu nekonzistentních, nepřesných nebo neplatných dat, které by mohlo mít negativní dopad na chod organizace. [1]

4 Vlastní práce

Výše rozebraná teoretická východiska k problematice optimalizace datových základen v relačně databázových evidencích budou v této kapitole ověřena na příkladu reálného podnikatelského subjektu s vhodnými parametry pro demonstraci hlavních úskalí nevhodné struktury datové základny.

Případová studie prezentovaná v této kapitole je řešena na základě výše popsaných teoretických předpokladů a autorových zkušeností s modelováním datových základen a správnou databází.

4.1 Popis modelové společnosti

Modelovým podnikatelským subjektem je zavedená a dynamicky se rozvíjející společnost zprostředkující komplexní reklamní služby od počáteční analýzy, přes zajištění či korekce grafického návrhu plakátů a výlepů, zajištění statistického šetření, zprostředkování tisku, až po výběr a zajištění vhodné inzertní plochy v požadovaných lokalitách po celé České republice. Společnost nabízí k pronájmu vlastní inzertní plochy, ale disponuje pouze velmi omezeným množstvím. Ve většině případech se tedy jedná čistě o poskytování zprostředkovatelských služeb. Personál společnosti čítá okolo deseti osob a skládá se především z obchodních zástupců, auditorů, grafiků a vývojářů aplikací. Průměrný roční obrat společnosti se pohybuje okolo 120 milionů korun.

4.2 Specifikace cílů a požadavků pro vytvoření databáze

Společnost v rámci budování vlastního informačního systému specifikovala požadavek na vytvoření jednotné datové základny pro tyto aplikační programy:

a) Interní aplikace pro evidenci a správu klíčových údajů
b) klientská webová aplikace

V rámci specifikace požadavků na interní aplikaci byly vymezeny tyto klíčové body:

a) Informace o zaměstnancích společnosti

b) Informace o obchodních partnerech a dodavatelích

c) Informace o klientech

d) Evidence inzertních ploch poskytovaných obchodními partnery

e) Evidence provedených auditů

f) Evidence zakázek

g) Evidence smluv

Cílem webové aplikace je poskytnout klientům jednoduchý, srozumitelný a graficky přívětivý přehled reklamní kampaně, který bude snadno dostupný z webového prohlížeče. Aplikace je koncipována jako mapa s postranním ovládacím panelem, pomocí kterého může uživatel sledovat zobrazovat či skrývat jednotlivé prvky dle vlastní potřeby. V rámci klientské aplikace je třeba uchovávat tyto údaje:

a) Přehled inzertních ploch včetně informací o lokaci

b) Přehled klientů i konkurenčních společností včetně informace lokaci o poboček

c) Evidence loga klientů i konkurence

d) Přehled zakázek

4.3 Sběr a analýza požadavků

Sběr požadavků a získávání podkladů pro případovou studii se opírá především o sérii rozhovorů s dvěma pověřenými pracovníky reklamní agentury. V prvním případě se jedná o majitele společnosti, který se zároveň podílel na vývoji interní aplikace pro správu zakázek, klientů a dodavatelů. Druhá osoba zastává ve společnosti nejen úlohu hlavního vývojáře aplikací, ale také databázového specialisty. V obou případech se jedná o kompetentní osoby se znalostí firemního „know-how“ a zároveň s hlubšími technickými znalostmi. Za účelem dosažení požadované míry poznání zkoumaného subjektu, hlubšího proniknutí do „workflow“ a zmapování aplikací organizace se studie opírá také o pozorování organizace za provozu a analýzu současné datové základny.

4.3.1 Požadavky specifikované zadavatelem

Na základě rozhovorů se zadavatelem byly zjištěny následující nedostatky stávající datové základky:

a) Nepřehledná struktura databáze

b) Náročný rozvoj a implementace nových segmentů datové základny

c) Konstrukce není přívětivá pro vývoj aplikačních programů

d) Výskyt nekonzistentních dat

Dále byly zadavatelem během provozu databáze zjištěny nedostatky, které je nezbytné při optimalizaci datové základny uvažovat. Na základě těchto zjištění byly definovány následující změnové požadavky:

a) Zlepšení evidence poskytovaných služeb

b) Vyřazení nevyužívaných prvků

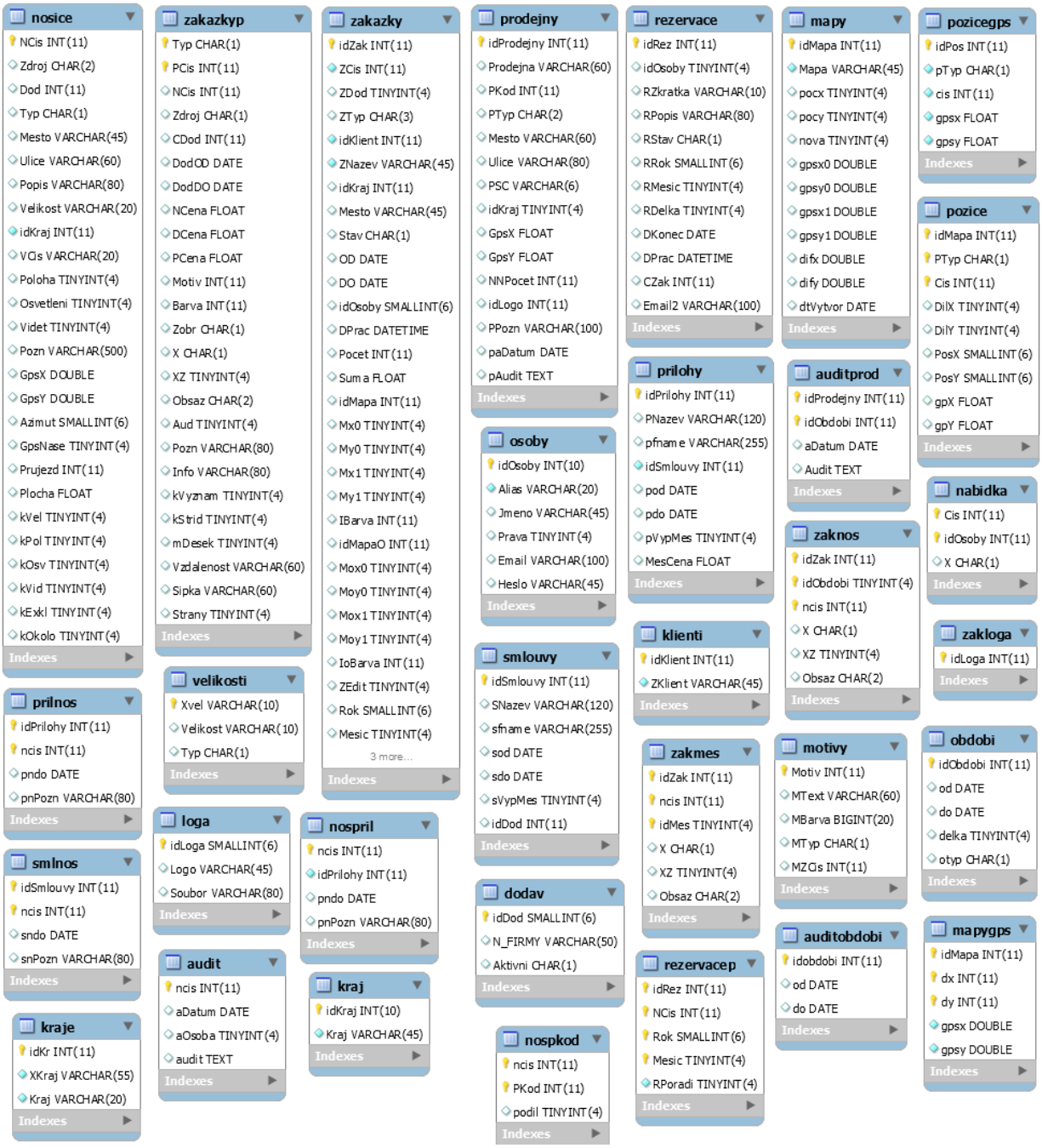
4.3.2 Analýza současného stavu

Z analýzy současné databáze vyplývají možné příčiny zadavatelem uvedených potíží. Na základě analýzy byly zjištěny následující nedostatky a nežádoucí prvky:

a) Existence nevyužívaných tabulek
b) Existence nevyužívaných atributů
c) Databáze není normalizovaná
d) Nevhodné vazby mezi tabulkami
e) Chybějící integritní omezení
f) Výskyt redundantních záznamů
g) Neodpovídají datové typy primárních a cizích klíčů
h) Chybějící konvence pojmenování tabulek a atributů

4.4 Rozbor stávající struktury databáze

Identifikované nedostatky stávající struktury relačně databázové evidence jsou v této kapitole detailně rozebrány za účelem lepšího porozumění modelové situaci.



Obrázek 5 - Výčet tabulek stávající databáze (zdroj: autor)

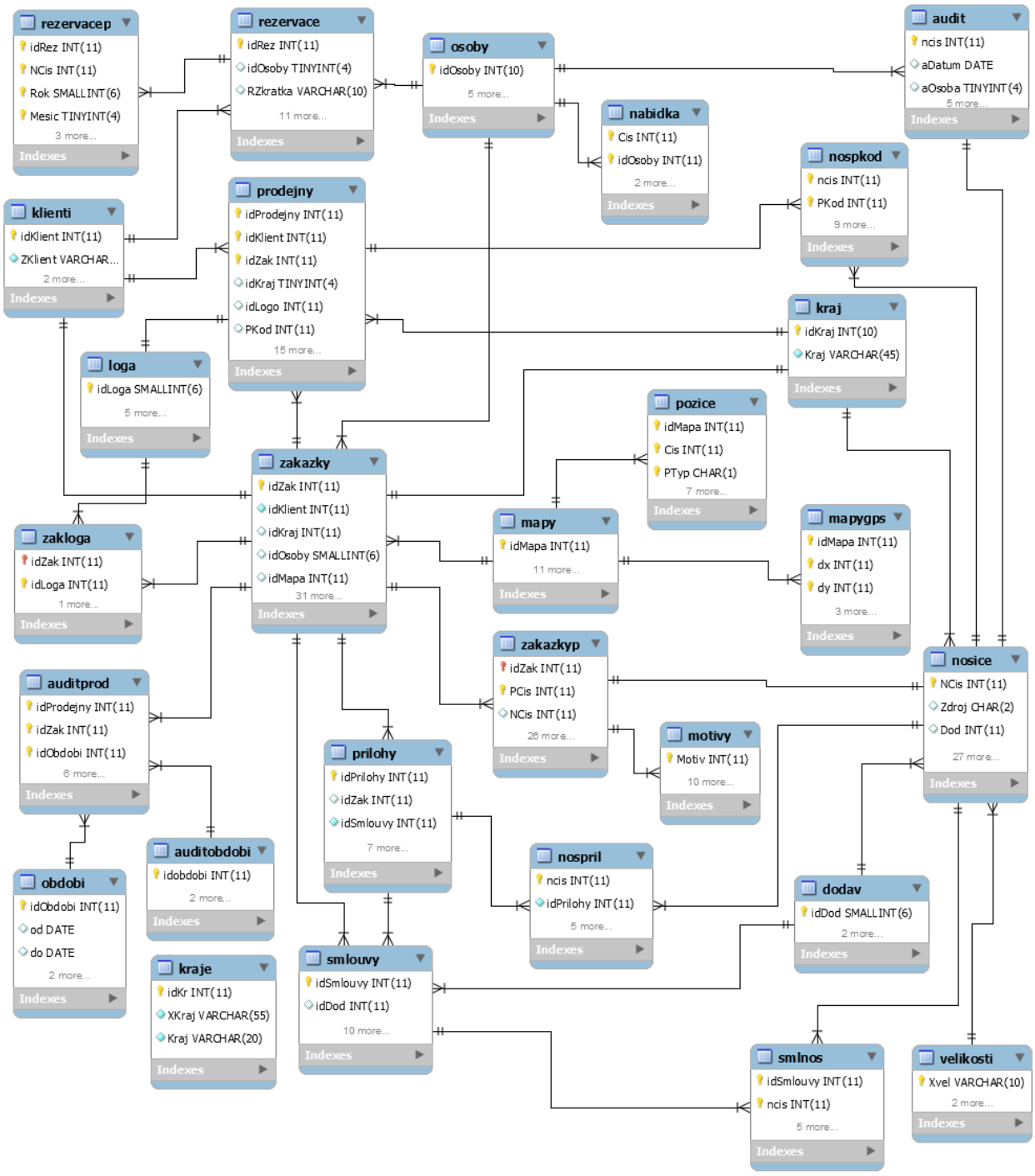
Diagram výše obsahuje úplný výčet tabulek stávající databáze včetně atributů a primárních klíčů. Přehled tabulek byl vygenerován v rozhraní MySQL Workbench pomocí nástroje „Reverse engineering“, který umožňuje na základě informací o současném schématu databáze vygenerovat UML diagram tříd. Správce databáze, vývojář či designér tak mohou jednoduše získat informace o struktuře datové základny v přehledné grafické podobě.

Zadavatel během série interview specifikoval řadu zastaralých tabulek, které se již v současné době nevyužívají. Jedná se o relace uvedené v tabulce níže, které nebudou při optimalizaci datové základny vůbec uvažovány.

Relace	**Typ relace**
mapy	datová tabulka
mapygps	datová tabulka
zakmes	vazební tabulka
zaknos	vazební tabulka
prilnos	vazební tabulka

Tabulka 1 – Nevyužívané relace (zdroj: autor)

V rámci databáze nejsou definována žádná integritní omezení, z toho důvodu nemohl nástroj „reverse engineering“ vygenerovat vztahy mezi tabulkami. MySQL Workbench umožňuje vygenerované schéma databáze editovat. Vzhledem ke komplexitě databáze je vhodné ze schématu za účelem zpřehlednění odebrat zastaralé tabulky, které se v současnosti nevyužívají. A následně do schématu manuálně doplnit stávající vazby mezi tabulkami. Manuálně upravený UML diagram je znázorněn na obrázku níže.



Obrázek 6 - Manuálně upravené schéma stávající databáze (zdroj: autor)

Na základě analýzy byla identifikována řada nežádoucích aspektů, z nichž některé mohou mít kritické následky. Především hrozí ztráta konzistence dat v důsledku nežádoucí redundance, absence číselníků a absence integritních omezení. Zároveň dochází k neúměrnému nárůstu objemu dat v databázi. Struktura databáze je v důsledku absence jmenné konvence objektů a nevhodně koncipovaných relací zbytečně složitá a nepřehledná. Je vhodné a žádoucí ji výrazně zredukovat.

Dle stávajícího procesu se záznam o inzertní ploše ukládá do databáze vždy s vytvořením nové zakázky obsahující tuto inzertní plochu, i v případě, že je již inzertní plocha v databázi evidována (pod jiným identifikačním číslem). V důsledku takovéto redundance roste velikost databáze a zároveň hrozí ztráta konzistence dat. Je žádoucí zajistit, aby byl o jedné inzertní ploše veden v databázi vždy pouze jeden záznam. V důsledku odstranění duplicitních záznamů je dále nutné upravit vazby s některými souvisejícími relacemi. Dopady jsou zobrazeny v tabulce níže.

Souvztažná relace	**Typ vztahu**	**vazební tabulka**	**Dopad**
kraj	1:M	---	Lze zachovat
zakazkyp	1:1	---	Nutno změnit
dovav	1:M	---	Lze zachovat
velikosti	1:M	---	Lze zachovat
smlouvy	M:N	smlnos	Lze zachovat
audit	1:1	---	Nutno změnit
prilohy	M:N	prilnos	Lze zachovat

Tabulka 2 – Dopad uvažované změny na asociace (zdroj: autor)

Stávající schéma obsahuje chybně stanovené vazby mezi některými relacemi. Zejména se jedná o vazby mezi jednotlivými segmenty objektu zakázka a vazby těchto segmentů na okolní relace. Nevhodně určená vazba byla identifikována také mezi relacemi *nosice* a *audit*. Designer stávající databáze vycházel z mylného předpokladu, že v rámci jedné zakázky proběhne vždy pouze jeden audit inzertní plochy. Na základě tohoto předpokladu stanovil typ vazby 1:1.

Relace *audit* a *auditprod* uchovávají data stejného charakteru v obdobné struktuře, rozdílem je pouze vazba na odlišné objekty. Relace *audit* se pojí k inzertním plochám, *auditprod* se váže k prodejnám. V tomto případě je vhodné relace sloučit. Sloučení je žádoucí také v případě dodavatelů a klientů. Relace obsahují v základu stejné údaje. Dodavatel může být zároveň i klientem, spojením těchto relací tedy lze zamezit případné redundanci. V rámci nově vzniklé relace lze evidovat rovněž požadované údaje o

konkurenčních subjektech. V návaznosti na sjednocení dat o podnikatelských subjektech lze do nově vzniklé tabulky přidat údaje z relace *loga*.

Relace *kraje* představuje číselník, který slouží ke konverzi nevhodně zadaných hodnot do vstupního formuláře na odpovídající název kraje. Zde by bylo žádoucí upravit vstupy na straně aplikace, která by umožňovala zadávat hodnoty pouze z číselníku.

Řada datových tabulek obsahuje atributy s předem definovanými hodnotami, které by bylo vhodné vyčlenit do číselníků. Užití číselníků přináší dvě velké výhody. V prvním případě číselníky usnadňují správu databáze. V budoucnu může nastat situace, kdy je třeba změnit hodnotu některé z evidovaných položek. Pokud se tato hodnota nachází pouze v číselníku, na který se příslušné tabulky odkazují pomocí cizích klíčů, je takováto změna velice jednoduchá a rychlá. V opačném případě může být tento proces značně obtížný. Vyčlenění do číselníku také redukuje obsah databáze, neboť v rozsáhlých tabulkách jsou uchovávány pouze číselné odkazy namísto dlouhých textových řetězců. Zejména se jedná o relace: *zakazky, zakazkyp, nosice, prodejny.*

Během analýzy bylo zjištěno, že některé relace obsahují atributy s hodnotami, které lze dopočítat. Jedná se zejména o relace: *zakazky, nosice, rezervace.*

Některé údaje jsou uchovávány ve dvou relacích současně. V důsledku toho roste objem databáze i riziko ztráty integrity dat. Jedná se o tyto záležitosti:

a) relace *auditprod* i *prodejny* obsahují informace o auditu

b) relace *rezervace* a *rezervacep* obsahují údaj o roku provedení rezervace

4.5 Návrh řešení

Během analýzy stávající databáze, firemních procesů a požadavků společnosti byly identifikovány a rozebrány nedostatky současné datové základny. V této části bude popsán proces optimalizace realizované za účelem odstranění výše vymezených nedostatků.

4.5.1 Jmenná konvence objektů

Obtížná orientace ve struktuře stávající datové základny je do značné míry způsobena nevhodným pojmenováním objektů v databázi. Absence jednotného rámce pro pojmenování objektů může zároveň znepříjemňovat psaní SQL dotazů. Nevhodně pojmenované objekty samozřejmě nepředstavují žádný prohřešek proti pravidlům relačního modelu dat. Tato zdánlivě zanedbatelná záležitost se však může v budoucnu, zejména v případě rozsáhlejších databází, vymstít. Z toho důvodu je vhodné vymezit alespoň základní rámec pravidel jednotného názvosloví.

Stanovení základních pravidel pro pojmenování objektů v databázi:

a) Pro pojmenování objektů bude užíván český jazyk bez diakritiky.

b) Pro pojmenování datových tabulek bude použito číslo jednotné a pascal case.

c) Pro pojmenování číselníků platí stejná pravidla jako pro datové tabulky. Název každého číselníku začíná prefixem „C_“ (za účelem snadného rozlišení).

d) Pro pojmenování vazebních tabulek platí stejná pravidla jako pro datové tabulky. Název vazební tabulky se skládá z kombinace názvů datových tabulek participujících na vazbě (případně z vhodné a jednoznačné zkratky). Název každého číselníku začíná prefixem „V_“ (za účelem snadného rozlišení).

e) Primární klíč je tvořen prefixem „ID_“ a názvem relace. Cizí klíče jsou tvořeny názvem rodičovské relace a postfixem „_ID“. Neklíčové atributy obsahují prefix jednoznačně určující jejich příslušnost k relaci, přičemž se může jednat buď o celý název relace, případně vhodnou zkratku, tento prefix je od popisné části oddělen podtržítkem. Počáteční slova v názvech atributů obsahují velké písmeno. V případě víceslovného názvu jsou jednotlivá slova oddělena podtržítkem. Atribut platnosti položky číselníku a data smazání prefix neobsahuje.

f) V ojedinělých a opodstatněných případech jsou připuštěny výjimky.

4.5.2 Konceptuální model

Z důvodu závažných strukturálních nedostatků, vzniklých v důsledku časové tísně při tvorbě stávajícího modelu, je vhodné absolvovat celý proces tvorby konceptuálního modelu. S ohledem na komplexitu modelované problematiky je aplikován postup modelování datové základny Shora dolů.

Na základě autorových zkušeností z oblasti vývoje software je pro tvorbu konceptuálního modelu zvolen standardizovaný modelovací nástroj UML 2.0. Pro tvorbu diagramů je použit počítačový program WhiteStarUML.

Na základě získaných podkladů byly v úzké spolupráci se zadavatelem vymezeny klíčové objekty, které je třeba v informačním systému podniku evidovat. Jedná se o následující prvky: dodavatelé, klienti, konkurenční subjekty, prodejny, smlouvy, zakázky, inzertní plochy, rezervace, audit, zaměstnanci. Na podkladě získaných poznatků o charakteru evidovaných dat lze již na této úrovni určit následující:

a) Zakázka může obsahovat více dílčích položek, které je nutné vyčlenit do samostatné relace. Evidované položky lze rozčlenit do dvou kategorií. V prvním případě se jedná o zprostředkování inzertní plochy. Druhá kategorie obsahuje všechny ostatní služby. Vazba zakázky na inzertní plochu bude realizována prostřednictvím relace *ZakazkaPolozka*, vazba na ostatní služby prostřednictvím relace *ZakazkaSluzba*.

b) Rezervace může obsahovat více dílčích položek, které je nutné vyčlenit do samostatné relace. Tato relace bude v konceptuálním modelu značena jako *RezervacePolozka*.

c) O klientech a dodavatelích jsou evidovány stejné údaje. Jeden objekt může zároveň vystupovat v roli klienta i dodavatele. Údaje o těchto subjektech je vhodné evidovat v rámci jedné relace, nazvané *Firma*. Zda se jedné o klienta nebo dodavatele bude rozlišeno na úrovni atributů.

d) Reklamní společnost chce evidovat údaje nejen o podnikatelských subjektech, se kterými spolupracuje, ale také o subjektech, které lze z pohledu klientů společnosti klasifikovat jako konkurenci (pro tyto subjekty bude dále užíván termín „konkurence“, nejedná se však o konkurenci z pohledu reklamní agentury). V případě konkurence je sledováno zejména rozmístění prodejen, případně reklamních materiálů, v dané lokalitě. Na základě uvedených předpokladů lze považovat za optimální řešení takovou alternativu,

kde jsou obecné údaje o všech subjektech, tedy o klientech, dodavatelích a konkurenci, evidovány v rámci entity *Firma* a detailní údaje o klientech a podnikatelích jsou vyčleněny do specializované entity *FirmaDetail*.

Diagram níže zobrazuje grafické znázornění předběžných entit v notaci UML. Pro přehlednost je již na úrovni konceptuálního modelu dodržována požadovaná jmenná konvence.

Obrázek 7 – Konceptuální model: Entity (zdroj: autor)

Nyní je třeba definovat vztahy mezi entitami se zřetelem na očekávaný způsob práce s daty. Definice vztahů včetně vhodného určení kardinality a participace zároveň vymezuje integritní omezení. Na základě získaných poznatků o informačním systému lze vymezit tyto vztahy:

a) Zakazka

V prvním kroku modelování byla entita reprezentující zakázky dekomponována do tří částí. První část obsahuje souhrnné informace o zakázce, ve zbylých částech jsou obsaženy jednotlivé položky. Zakázka se může skládat z jedné nebo více položek (*ZakazkaSluzba* a *ZakazkaPolozka*), každá položka náleží právě jedné zakázce.

Ke každé zakázce je evidován právě jeden zaměstnanec, který ji vytvořil. Zaměstnanec může vytvořit více zakázek, ale také nemusí vytvořit zakázku žádnou.

Zakázka se vztahuje právě k jednomu klientovi. K firmě se může vázat více zakázek. Zároveň je nutné uvažovat, že v tabulce firem jsou evidovány i jiné entity než klienti, ke kterým se nebude pojit žádná zakázka.

K vyjasnění vztahu mezi entitami *Zakazka* a *Smlouva* je nejprve nutné vymezit, co *Smlouva* v modelu reprezentuje. Obsahem tohoto objektu je cesta k umístění naskenovaných smluv na disku. Ke každé zakázce se standardně váže jedna smlouva, může ale nastat situace, kdy jsou ke smlouvě přidávány dodatečné dokumenty, které jsou poté uloženy na disku jako samostatné soubory. Ačkoli se nemusí jednat o smlouvu v pravém slova smyslu, v tabulce se objeví jako samostatný záznam. Z toho vyplývá, že se k jedné zakázce může vázat více záznamů z relace *Smlouva*. Zde je však nutné brát v potaz časovou prodlevu mezi vytvořením zakázky v informačním systému a naskenováním smlouvy. Zakázky v databázi tedy existují zpravidla určitou dobu beze smlouvy. Smlouva se oproti tomu váže vždy jen k jedné zakázce. Toto platí v případě, že se jedná o smlouvu s klientem. V tabulce smluv se nacházejí také smlouvy s dodavateli, které se váží k relaci *ZakazkaPolozka* nikoli *Zakazka*.

b) *ZakazkaSluzba* a *ZakazkaPolozka*

K položkám zakázky se pojí jednotliví dodavatelé. *ZakazkaSluzba* i *ZakazkaPolozka* jsou tedy ve vztahu s relací *Firma*. Zde je důležité připomenout, že *Firma* neobsahuje pouze dodavatele. Dále je nutné počítat s faktem, že reklamní agentura je schopna poskytnout některé služby sama. K položce zakázky se tedy váže jedna nebo žádná firma. K firmě se může vázat více položek zakázky, ale také se k ní nemusí vázat položka žádná.

K položce zakázky se může vázat maximálně jedna smlouva (s dodavatelem). Jedna smlouva ale může pokrývat více položek.

K relaci *ZakazkaPolozka* se navíc ještě pojí inzertní plocha (entita *nosic*). Evidovaná inzertní plocha může figurovat ve více položkách nebo nefiguruje v žádné položce. Položka standardně obsahuje jednu inzertní plochu, ale ve specifických případech nemusí být položka vázána k žádné ploše.

c) *Firma*

Všechny podnikatelské subjekty, o kterých potřebuje reklamní agentura uchovávat údaje, jsou reprezentovány pomocí relace *Firma*. Případné detailnější informace o určitých subjektech jsou vyčleněny do relace *FirmaDetail*. Detail náleží právě jedné firmě. O firmě může nebo nemusí být evidován detail.

Klient (firma) nebo konkurenční subjekt (také firma) může disponovat libovolným množstvím prodejen, jejichž poloha je v obou případech důležitá z hlediska alokace reklamy. Prodejna se vždy váže právě k jedné firmě.

Klient (firma) může vytvořit libovolný počet rezervací. Rezervace se vždy pojí pouze s jedním klientem.

Dodavatel (firma) může disponovat inzertními plochami. Inzertní plocha se vždy pojí právě k jednomu dodavateli.

d) *Nosic*

Reklamní plocha bývá často v rámci zakázky spjata s konkrétní prodejnou klienta. Tento vztah je třeba v informačním systému evidovat. Inzertní plocha je k prodejně vázána vždy pouze v určitém časovém období. Z důvodu zajištění časového rozlišení nejsou entity *Nosic* a *Prodejna* propojeny napřímo, ale přes tabulku *NosicAlokace,* kde je období vymezeno. *NosicAlokace* zároveň plní účel vazební tabulky ve vztahu M:N. Reklamní plocha může být v čase vázána na libovolné množství prodejen. Prodejna může být (i v jednom období) vázána na více ploch.

Reklamní plocha může být dále ve vztahu s entitou *Audit*. Během auditu může být prověřeno libovolné množství ploch. Vzhledem k tomu, že lze kromě inzertních ploch auditovat i prodejny, nemusí se Audit vázat k žádné inzertní ploše. Reklamní plocha může být auditována opakovaně. Zde je časový údaj evidován přímo v rámci tabulky *Audit*.

Klientovi může být reklamní plocha rezervována. V takovém případě se plocha váže k položce rezervace (*RezervacePolozka)*. Nosič může mít vazbu na libovolný počet položek. Pochopitelně by neměl být jeden nosič vázán k více položkám ve stejném časovém období. Časové rozlišení je evidováno v relaci *RezervacePolozka*. Každá položka rezervace je vázána k právě jedné inzertní ploše.

e) *Rezervace*

Výše zmíněná rezervace inzertní plochy je v rámci modelu rozdělena do dvou částí. Rezervace se může skládat z více položek (může obsahovat více inzertních ploch), z toho důvodu jsou nosiče vázány k jednotlivým položkám rezervace. Položky jsou sdruženy v rámci relace *Rezervace*. Každá položka náleží právě k jednomu záznamu v tabulce *Rezervace*. K rezervaci se může vázat libovolné množství položek, minimálně však jedna.

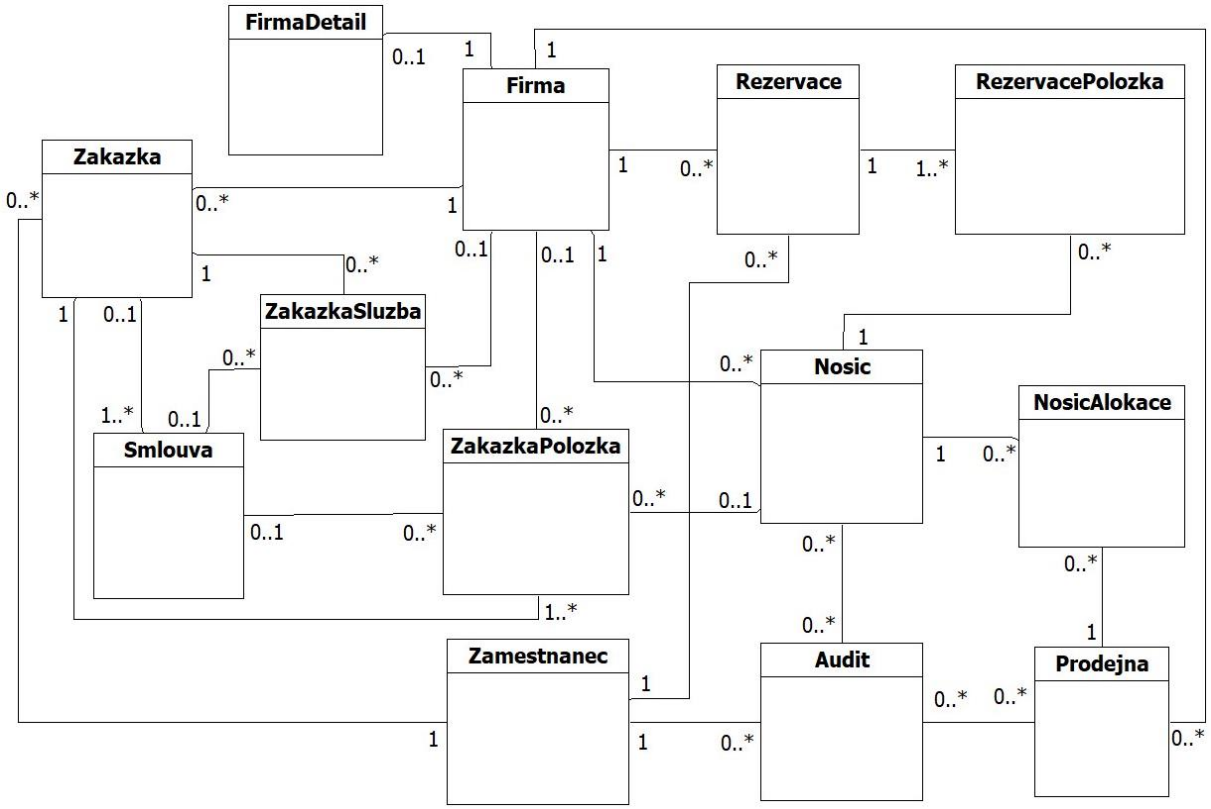
Rezervace se dále váže ke klientovi, jak bylo vymezeno výše. Kromě toho je u každé rezervace požadováno evidovat záznam o zaměstnanci, který ji vytvořil. Ke každému záznamu z této entity je tedy vázán právě jeden zaměstnanec. Zaměstnanec může vytvořit libovolné množství rezervací.

f) *Audit*

Kromě auditu reklamní plochy je možné auditovat také prodejny, na které lze rovněž umisťovat reklamní materiál. K auditu může být vázáno libovolné množství prodejen (nemusí k němu být vázána žádná prodejna). K prodejně se může pojit libovolné množství auditů.

Ke každému auditu se navíc váže záznam o auditorovi (zaměstnanci). Audit je tedy spojen právě s jedním zaměstnancem. K zaměstnanci se může vykonat libovolné množství auditů.

Výše uvedená tvrzení jednoznačně určují kardinalitu a participaci vazeb mezi všemi objekty modelované reality. Diagram níže zobrazuje rozpracovaný konceptuální model v notaci UML doplněný právě o výše popsané asociace.



Obrázek 8 – Konceptuální model: Asociace (zdroj: autor)

Dalším krokem modelování je doplnění atributů do předběžných relací. Během tohoto procesu dochází k dekompozici objektů v důsledku vhodného vyčlenění určité podmnožiny atributů do samostatné relace. Nejčastěji se jedná o případy, kdy je požadováno k jednomu záznamu evidovat více údajů stejného charakteru, přičemž množství se může u jednotlivých záznamů značně lišit. V tomto případě byly již v první fázi modelování některé entity s ohledem na charakter dat předběžně rozděleny na více souvisejících tabulek. Zároveň jsou během této fáze některé atributy se specifickými hodnotami vyčleněny do číselníků. Vyčlenění atributů do číselníků je dáno množinou hodnot, kterých mohou nabývat. Proto je vhodné zabývat se charakterem dat již na této úrovni modelování.

Během doplňování atributů do předběžných relací byly identifikovány prvky, které mohou nabývat pouze hodnoty z předem vymezené množiny. Pro takovéto prvky je žádoucí vytvořit číselníky obsahující danou množinu hodnot. V modelu jsou užity dvě struktury číselníků v závislosti na charakteru obsahu, především na očekávané frekvenci jeho změn. V případě množiny hodnot, u které lze očekávat možné budoucí změny obsahuje číselník příznak, zda byla hodnota smazána a datum smazání. Tímto způsobem lze zajistit, že v případě starších záznamů zůstane tato hodnota v evidenci zachována, ale zaměstnancům se již v aplikaci nezobrazí. V případě číselníků, kde je změna hodnot velice nepravděpodobná nebo se jedná o dynamické ukazatele (například stavy), se tento příznak nepoužívá. Tabulka níže zobrazuje přehled číselníků.

Údaj	**Číselník**	**Příznak**	**Poznámka**
Typ zakázky	C_TypZakazky	Ano	Lze očekávat změny obsahu číselníku.
Typ služby	C_TypSluzby	Ano	Lze očekávat změny obsahu číselníku.
Typ prodejny	C_TypProdejny	Ano	Lze očekávat změny obsahu číselníku.
Typ nosiče	C_TypNosice	Ano	Lze očekávat změny obsahu číselníku.
Zdroj nosiče	C_Zdroj	Ano	Lze očekávat změny obsahu číselníku.
Zobrazeni	C_Zobrazeni	Ano	Lze očekávat změny obsahu číselníku.
Stav zakázky	C_StavZakazky	Ne	Jedná se o stav.
Stav rezervace	C_StavRezervace	Ne	Jedná se o stav.
Tabulka krajů	C_Kraje	Ne	Nejsou předpokládány změny krajů.

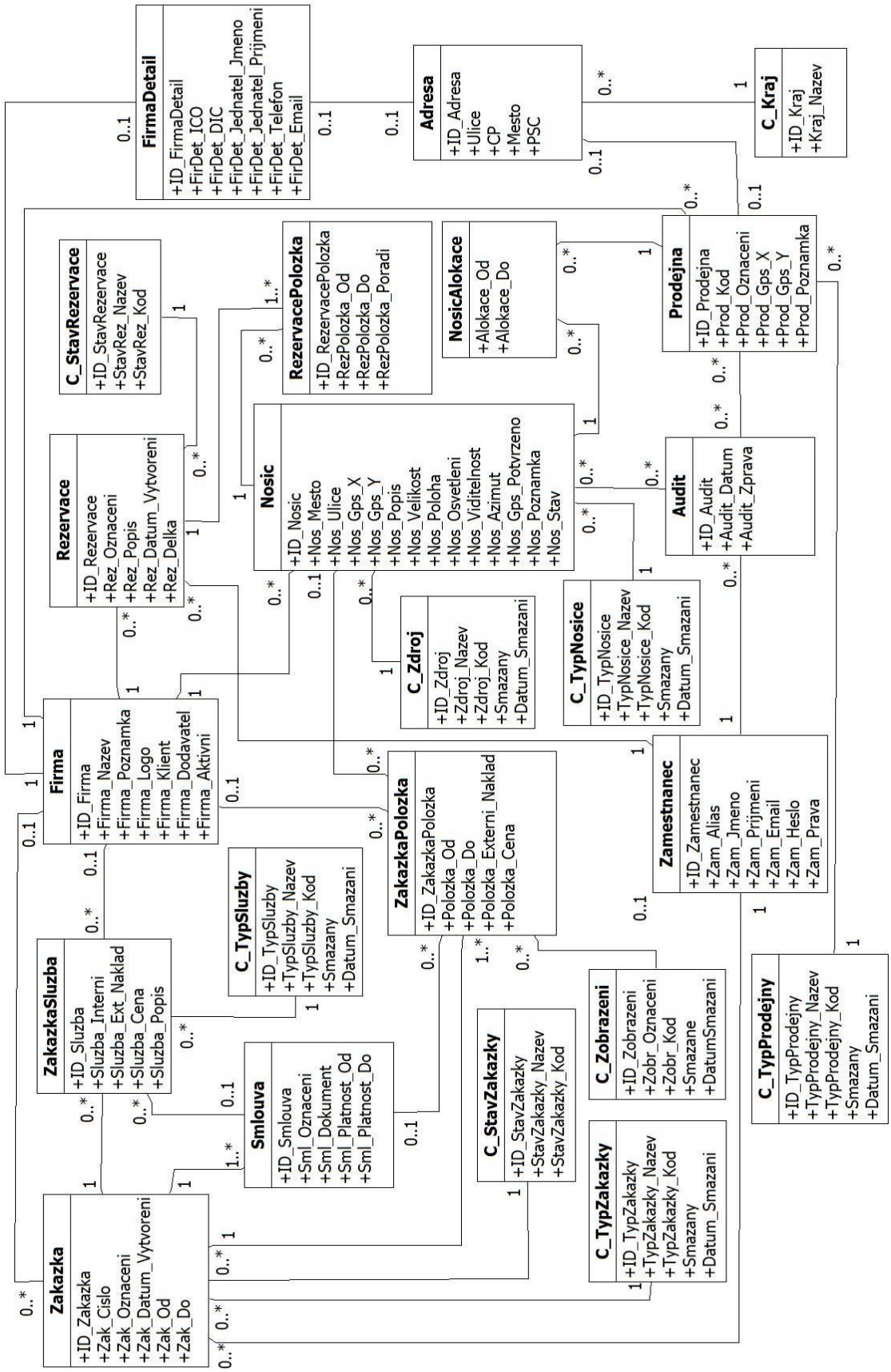
Tabulka 3 – Přehled číselníků (zdroj: autor)

Trh se vyvíjí v čase, rovněž u reklamní agentury lze v tomto ohledu předpokládat vývoj v čase. Proto je třeba brát v potaz například změnu složení poskytovaných služeb v závislosti na vývoji trhu. Agentura může rozšířit nabídku o nové služby, ale zároveň může přestat poskytovat některé nerentabilní služby. Obdobné změny platí například i pro typ zakázky nebo typ prodejny. Oproti tomu lze předpokládat, že změna vyšších územních samosprávních celků na území České republiky není v horizontu blízké budoucnosti příliš pravděpodobná.

Specifickou skupinu pak představují stavy. Jedná se o dynamické ukazatele, které se mění v čase. V tomto případě je sice pravděpodobnost změny obsahu číselníků vyšší než u krajů, na druhou stranu ale není žádoucí, aby se záznam nacházel v neexistujícím stavu. Vznikne-li například požadavek na změnu názvu stavu, stačí upravit pouze požadovanou položku číselníku. V případě požadavku na úplné smazání některého stavu je nutné nejprve aktualizovat všechny záznamy obsahující referenci na tento vztah.

Kromě výše uvedených číselníků byla z relací *FirmaDetail* a *Prodejna* vyčleněna adresa. V rámci informačního systému je požadováno evidovat adresu sídla firmy, korespondenční adresu a případně adresy prodejen. Reklamní agentura spolupracuje s řadou menších subjektů, u kterých jsou všechny tyto adresy shodné. Vyčleněním adresy do samostatné relace lze zajistit, aby byl v takovém případě v databázi uložen pouze jeden záznam. Ačkoli tento krok nepřinese při současné velikosti databáze výrazné snížení objemu uložených dat, jedná se o užitečné opatření snižující riziko ztráty datové integrity.

Diagram níže reprezentuje konceptuální model s doplněnými atributy a číselníky, včetně výše zmiňovaného vyčlenění adresy.



Obrázek 9 – Konceptuální model: Atributy (zdroj: autor)

Posledním krokem konceptuálního modelování je v tomto případě určení primárních klíčů, tedy atributů jednoznačně identifikujících každý výskyt entity. Lze předpokládat, že položky v rámci číselníků budou unikátní. Kromě oficiálních názvů položek obsahuje navíc každý číselník ještě kód, který by měl být rovněž unikátní. Oba atributy tedy lze považovat za kandidátní klíče. Z pohledu zpracování dotazů je ale vhodné odkazovat se na položky číselníků prostřednictvím číselné hodnoty. Z tohoto důvodu jsou do číselníků přidány ještě číselné identifikátory označené prefixem ID. Vzhledem k vysoké provázanosti jednotlivých entit je doplněn číselní identifikátor do všech tabulek s výjimkou tabulky *NosicAlokace*, která zde figuruje v roli vazební tabulky mezi inzertními plochami a prodejnami. V případě vazební tabulky *NosicAlokace* je dána jedinečnost záznamu všemi atributy. U všech ostatních tabulek je z důvodu rychlosti vyhodnocování dotazů zvolen za primární klíč právě číselní identifikátor.

Z jednotlivých kroků konceptuálního modelování je zřejmé, že obecná posloupnost úkonů vymezená ve výše popsané metodologii návrhu a optimalizace datové základny relačně databázové evidence nebyla během návrhu striktně dodržována. Zde je na místě připomenout, že sama metodologie připouští odchylky v závislosti na řešené problematice. Vzhledem k autorovým zkušenostem a míře zmapování současné situace, ke které napomohla i velice dobrá komunikace se zadavatelem, bylo možné některé kroky úmyslně do schématu zakomponovat již ve dřívějších fázích návrhu. V důsledku toho byl proces modelování zkrácen o řadu iterací. Ačkoli posloupnost některých úkonů přímo nekorespondovala s obecnou šablonou, byly vykonány všechny kroky modelovacího procesu, čímž byla zajištěna kontrola výchozích předpokladů.

4.5.3 Relační model dat

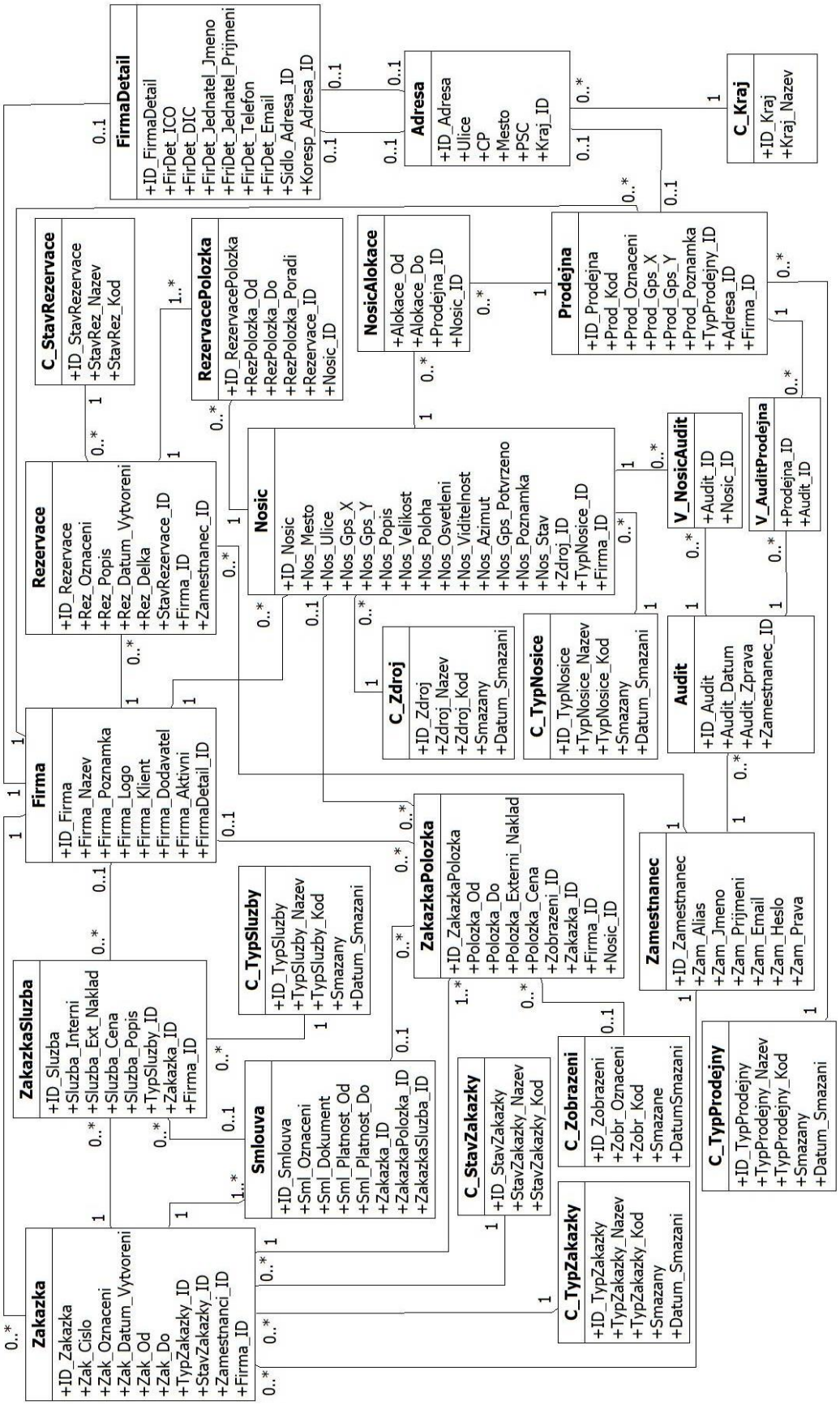
Konceptuální schéma je nyní třeba transformovat do struktury odpovídající relačnímu modelu dat. Během procesu transformace by mělo být schéma restrukturalizováno tak, aby výsledný relační model neobsahoval složené ani vícehodnotové atributy. Na této úrovni jsou zároveň realizovány odpovídající vazby mezi relacemi.

Ve tomto specifickém případě se autor zabýval vyčleněním vícehodnotových atributů již na konceptuální úrovni návrhu. V řadě případů by mohlo být nevhodné nebo dokonce nežádoucí zatěžovat konceptuální model podobnými aspekty, neboť tento model slouží především jako jednoduchý a srozumitelný komunikační kanál mezi zadavatelem a designérem. V případě reklamní agentury však designer komunikoval s technicky vzdělanými zaměstnanci, kterým je relační model dat blízký. Na základě těchto specifických skutečností považuje autor za optimální zakomponovat prvky predikovatelné téměř s jistotou již v dřívějších fázích modelování.

Na základě prověření konceptuálního modelu lze prohlásit, že schéma neobsahuje žádné entity se složenými ani vícehodnotovými atributy. Z tohoto pohledu nedojde při transformaci na logický návrh k žádným strukturálním změnám v důsledku dekompozice atributů ani jejich vyčleněním do samostatných tabulek. V rámci přiřazování atributů do konceptuálního návrhu došlo zároveň k vyčlenění specifických atributů do číselníků.

Nedílnou součástí transformace je již zmiňovaná realizace vazeb mezi jednotlivými relacemi. V relačním modelu jsou vazby mezi záznamy realizovány prostřednictvím cizích klíčů. Vztahy mezi objekty byly identifikovány a popsány již na úrovni konceptuálního modelu. U každého vztahu byla vymezena kardinalita a participace. Na základě těchto poznatků lze v souladu s pravidly vymezenými v teoretických východiscích doplnit do příslušných relací cizí klíče, případně doplnit do schématu vazební tabulky.

Diagram níže zobrazuje schéma v notaci UML upravené do podoby korespondující s relačním datovým modelem. U binárních vztahů typu 1:N a 1:1 (s nepovinnou participací na jedné straně) jsou do tabulek doplněny cizí klíče. V případě binárních relací typu M:N je schéma rozšířeno o vazební tabulky.



Obrázek 10 – Relační model (zdroj: autor)

4.5.4 Normalizace

V této fázi modelování je žádoucí prověřit, zda navržená struktura relačního modelu odpovídá kritériím požadované normální formy. Dle obecného předpokladu z praxe je optimální prověřovat, zda model splňuje třetí normální formu. V tomto případě navíc navržené řešení neobsahuje žádné složené primární klíče. V případě splnění 3NF lze tvrdit, že je model zároveň v souladu s pátou, tedy nejvyšší, normální formou.

V případě zkoumání první normální formy zřetelně vyplývá na povrch výhoda dodržení metodologie návrhu a optimalizace datové základny. Zajištění nedělitelnosti hodnot bylo prověřeno již při transformaci konceptuálního schématu na logický návrh. Na základě přezkoumání této záležitosti v rámci prověřování pravidel normalizace lze konstatovat, že navržené schéma skutečně odpovídá první normální formě.

Žádná relace ve schématu neobsahuje složený primární klíč. V tomto případě lze bez dalšího zkoumání tvrdit, že je schéma v souladu s podmínkami druhé normální formy.

Třetí normální forma se zabývá tranzitivními závislostmi uvnitř relace. V rámci prověřování schématu na základě kritérií této normální formy je třeba věnovat pozornost zejména tabulce *Adresa*, která obsahuje mimo jiné atributy *Mesto* (město) a *PSC* (poštovní směrovací číslo). V České republice vztah mezi městem a PSČ bezpochyby existuje. V rámci informačního systému reklamní agentury však může nastat situace, kdy jsou v databázi evidovány dva rozdílné textové řetězce reprezentující jedno město. Je to dáno tím, že zaměstnanci zapisují adresu ručně do textového pole. Jistě je na místě zvážit přidání komplexního číselníku měst a PSČ. Takovýto číselník s řádově tisíci záznamy je navíc veřejně dostupný a není nutné ho vytvářet. Objem číselníkové tabulky je ale neúměrně velký reálně využívaným záznamům. Zařazení číselníku by navíc vyžadovalo implementaci změn na straně aplikačního programu. Přičemž lze předpokládat, že přidání listboxu s tisíci položkami by jistě nebylo vyhovujícím řešením. Dále se nabízí řešení prostřednictvím dynamického číselníku, který by byl v případě potřeby doplňován zaměstnanci, tím by byla zajištěna relevantnost i přijatelné množství záznamů. I toto řešení by ale vyžadovalo implementaci změn na straně aplikačního programu, přičemž by se z pohledu současného workflow nemuselo jednat o změnu žádoucí. S ohledem na zmiňované faktory lze považovat stávající řešení za optimální. Zároveň lze tvrdit, že datový model splňuje požadovanou úroveň normalizace.

4.5.5 Integritní omezení

Pravděpodobně největší úskalí stávající databáze představuje ztráta nekonzistence dat. K udržení konzistence z velké části napomáhá normalizace struktury databáze. Je však vhodné věnovat dostatek pozornosti také vymezení pravidel pro aktualizaci dat.

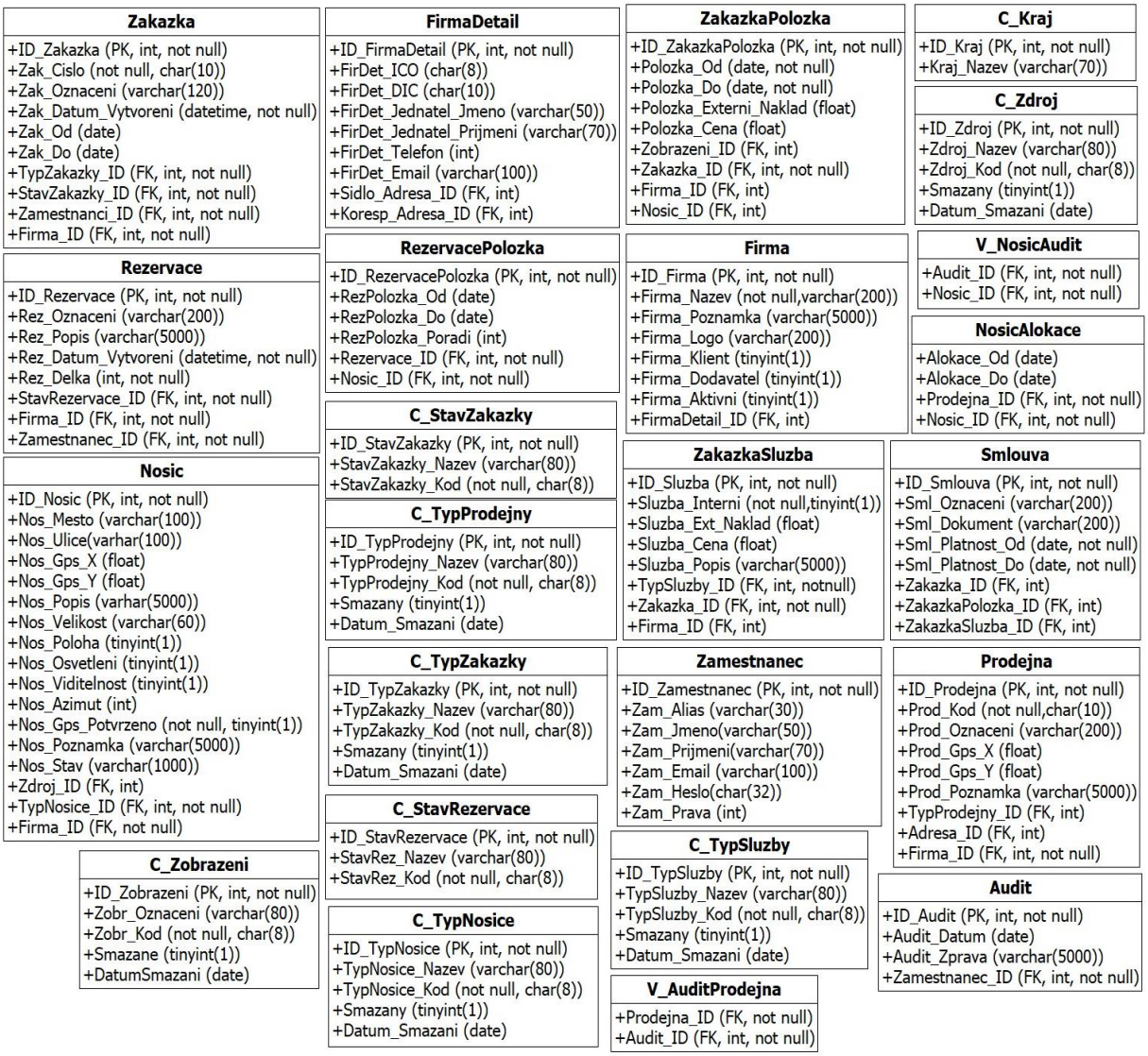
Pravidla pro referenční integritu byla již formulována v rámci návrhu struktury databáze. Entitní integritu je SŘBD schopen zajistit pomocí kontroly jedinečnosti primárního klíče. Je však nutné vymezit ještě doménovou integritu.

Mechanismy zajišťující zachování integrity v případě mazání záznamů jsou v současné době implementovány na straně aplikačního programu. Samotné mazání položek je předpokládáno pouze ve specifických a ojedinělých případech. Přímý přístup do databáze má pouze pověřený zkušený pracovník, který je zde navíc částečně hlídán omezením referenční integrity. Na základně těchto skutečností lze považovat současnou implementaci kontrolních mechanismů na straně aplikací za dostatečnou a absenci procedur na straně databáze za akceptovatelnou. Diagram níže zobrazuje relace doplněné o doménovou integritu.

Níže uvedený diagram tříd znázorňuje vymezení domén. Z důvodu přehlednosti zde nejsou zobrazeny vazby mezi relacemi. Užité značení datových typů je v souladu s databází MySQL. Přehled použitých hesel je uveden v tabulce níže.

Zkratka	Anglický název	Český popis	Upřesnění
PK	primary key	primární klíč	
FK	foreing key	cizí klíč	
int	integer	celé číslo	
float	float	desetinné číslo	
char	character	textové pole	parametr v kulaté závorce udává počet znaků
varchar	variable character	textové pole	parametr v kulaté závorce udává max. počet znaků
tinyint	tiny integer	celé číslo	s parametrem 1 reprezentuje boolean hodnotu
date	date	datum	
datetime	datetime	datum a čas	
not null	not null	neprázdný	pole musí být vyplněno

Tabulka 4 – Značení datových typů (zdroj: autor)



Obrázek 11 - Vymezení domén (zdroj: autor)

5 Závěr

Hlavním cílem práce byla optimalizace struktury datové základny za účelem zabezpečení optimální práce s daty, zajištění datové integrity a zvýšení robustnosti databázového systému.

V první části práce byla na základě studia dostupných informačních zdrojů v oblastech relační databázové technologie, metodologie vývoje relačních databází a datového modelování formulována podstatná teoretická východiska problematiky optimalizace struktury relačně koncipovaných datových základen. Rovněž zde byla vymezena základní terminologie konceptuálního modelování a relačního modelu dat. V závěru teoretické části byl rozebrán proces optimalizace struktury datové základy.

V praktické části byla řešena reálná situace z podnikové praxe. Nejprve byla specifikována základní charakteristika zkoumaného podnikatelského subjektu, který se potýká s nedostatky nevhodně koncipované datové základny. Následně byla provedena analýza současného databázového systému a sběr požadavků od zadavatele. Tato fáze se opírala především o rozhovory se zadavatelem, pozorování organizace za provozu a zkoumání současné struktury databáze.

Během fáze sběru a analýzy podkladů byly identifikovány strukturální nedostatky současné datové základny, z nichž k nejzávažnějším patří výskyt redundantních záznamů a absence integritních omezení v důsledku čehož dochází ke ztrátě konzistence evidovaných dat. Dále bylo zjištěno, že zkoumaná datová základna není normalizovaná, obsahuje nevhodné vazby mezi některými relacemi a v určitých případech doména cizího klíče nekoresponduje s doménou odpovídajícího primárního klíče. Mezi méně závažné nedostatky byla následně zařazena absence jednotné konvence pro pojmenování objektů v databázi a výskyt nevyužívaných tabulek i atributů.

Na podkladě informací získaných během analytické fáze byl následně realizován návrh odpovídajícího řešení s ohledem na požadavky a potřeby organizace. V první řadě byla stanovena konvence pro pojmenovávání objektů v databázi. V souladu s metodologií byl prostřednictvím UML diagramu tříd v úzké spolupráci se zadavatelem následně konstruován konceptuální model datové základy. Vzniklé schéma bylo transformováno do struktury relačního modelu dat. V další fázi bylo zkoumáno, zda logický návrh splňuje podmínky třetí normální formy. Dodržení procesu návrhu, respektive optimalizace, datové

základny výrazně napomohlo tomu, že výsledné schéma na logické úrovni splnilo podmínky požadované normální formy, tudíž nebylo třeba provádět další dekompozici a restrukturalizaci schématu. V závěru byla navržena jednotlivá integritní omezení.

Výsledkem práce je návrh optimalizované struktury databáze včetně vymezení integritních omezení, doporučení procesní změny a návrhu změn, které je nutné implementovat na straně aplikačního programu. Navržená struktura je v důsledku odstranění nevyužívaných objektů a zavedení jednotného názvosloví přehlednější, což usnadňuje administraci databáze, vývoj aplikačních programů i analýzu dat. V důsledku restrukturalizace databáze do normalizované podoby, vyčlenění specifických hodnot do číselníků a využití hierarchie je datová základna flexibilnější vůči možným budoucím změnám, zároveň je zajištěna snazší práce s daty a rychlejší odezva databáze. Odstraněním redundantních hodnot a vyčleněním hodnot do číselníků lze docílit také redukce objemu databáze. Nejdůležitějším aspektem však je, že optimalizovaná struktura spolu s navrženými integritními omezeními minimalizuje riziko ztráty konzistence a integrity dat.

Přes zmiňovanou nutnost individuálního přístupu k řešení této problematiky lze některé důležité aspekty zobecnit pro další použití. Během návrhu optimální struktury databáze se jednoznačně potvrdila důležitost zajištění adekvátní komunikační vazby mezi designérem a kompetentními zástupci organizace v průběhu celého procesu. Dále vyplynul na povrch pozitivní dopad použití vhodné metodologie. Ačkoli reálný rozdíl v objemu evidovaných dat i rychlosti zpracování SQL dotazů (zejména nad velkým množstvím záznamů), bude možné vyčíslit až po nasazení modelu do provozu, pozitivní tendence je patrná již nyní. Na základě toho lze potvrdit pozitivní vliv normalizovaného schématu na chod databázového systému. Normalizovaná struktura spolu s navrženými integritními omezeními se viditelně podílí také na minimalizaci rizika ztráty konzistence evidovaných dat. Z práce je rovněž patrná analogie procesu optimalizace logického schématu databáze s návrhem nové datové základy, která je dána nutností oprostit se od chybných úvah a předpokladů, na jejichž základě byla původní struktura databáze navržena.

Závěrem lze zdůraznit podstatný vliv struktury datové základny na fungování celého informačního systému. Především případná ztráta dat, respektive ztráta datové integrity, v důsledku nevhodného návrhu může mít na provoz organizace velice negativní dopad. Logickým návrhem databáze je tedy nutné se zabývat.

6 Seznam použitých zdrojů

1. BEGG, C., CONOLLY, T., HOLOWCZAK, R.: Mistrovství databáze, profesionální průvodce tvorbou efektivních databází. Computer Press. Brno 2009. ISSBN 978-80-251-2328-7
2. HERMANDEZ, M.: Návrh databází, GRADA 2005. ISBN 80-247-0900-7
3. TEOREY, Toby J. Database modeling and design: logical design. 5th ed. Burlington: Morgan Kaufmann, c2011. The Morgan Kaufmann series in data management systems. ISBN 978-0-12-382020-4.
4. GARMANY, John, Jeff WALKER a Terry CLARK. Logical database design principles. Boca Raton: Auerbach, 2005. Foundations of database design series. ISBN 0-8493-1853-X.
5. POKORNÝ, Jaroslav. Databázové systémy 2. Praha: Nakladatelství ČVUT, 2007. ISBN 978-80-01-03797-3.
6. Vztahy mezi objekty a třídami: Agregace a kompozice [online]. [cit. 2017-08-20]. Dostupné z: http://www.cs.vsb.cz/benes/vyuka/upr/texty/objekty/index.html
7. Creating Databases [online]. [cit. 2017-07-17]. Dostupné z: http://www2.amk.fi/digma.fi/www.amk.fi/opintojaksot/0303011/1146161367915.html
8. INTRODUCTION TO OMG'S UNIFIED MODELING LANGUAGE [online]. 2005 [cit. 2017-08-02]. Dostupné z: http://www.uml.org/what-is-uml.htm
9. Database Normalization Basics [online]. 2017 [cit. 2017-08-09]. Dostupné z: https://www.thoughtco.com/database-normalization-basics-1019735
10. Generalizace/specializace [online]. 2011 [cit. 2017-08-12]. Dostupné z: http://krokodata.vse.cz/DM/ISA