

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

Navrhování datové základny databázových evidencí

Heřmánek Jan

© 2015 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Katedra informačního inženýrství

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Jan Heřmánek

Informatika

Název práce

Navrhování datové základny databázových evidencí

Název anglicky

Projecting a data base of database records

Cíle práce

Bakalářská práce je tematicky zaměřená na problematiku navrhování datových základen relačních databázových evidencí. Cílem práce je:

- vymezit teoretické principy problematiky relačních databází, navrhování jejich datových základen včetně jejich datové normalizace,
- zmapovat současnou situaci v této záležitosti, identifikovat požadavky kladené na takovéto evidence se zřetelem na eliminaci případných datových redundancí,
- navrhnout a následně ověřit možnosti řešení těchto požadavků na konkrétním příkladu z praxe,
- ověřené záležitosti zobecnit pro další možná použití.

Metodika

Použitá metodika zadané bakalářské práce bude založena na studiu a analýze dostupných informačních zdrojů a existujících řešení v dané oblasti. Stěžejní pro vypracování této závěrečné práce budou metody a techniky relačně databázové technologie v kontextu s modelováním datové základny. 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.

Doporučený rozsah práce

40-50 stran

Doporučené zdroje informací

BEGG, C., CONOLLY, T., HOLOWCZAK, R.: Mistrovství databáze, profesionální průvodce tvorbou efektivních databází. Computer Press. Brno 2009. ISSN 978-80-251-2328-7
BRYLA, B., LONEY, K.: Mistrovství v Oracle Database 10g. Computer Press Brno 2006. EAN 978802512779
HERMANDEZ, M.: Návrh databází, GRADA 2005. ISBN 80-247-0900-7.
LACKO, L.: ORACLE. Správa, programování a použití databázového systému. Computer Press Brno 2007. EAN 97880251149002.
LONEY, K.: Oracle Database, kompletní průvodce. Computer press Brno 2010. ISBN 978-80-251-2489-5
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

Předběžný termín obhajoby

2015/06 (červen)

Vedoucí práce

doc. Dr. Ing. Václav Vostrovský, Ph.D.

Elektronicky schváleno dne 10. 11. 2014

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 10. 11. 2014

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 11. 03. 2015

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Navrhování datové základny databázových evidencí" 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 literatury 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 11.3.2015

Poděkování

Rád bych touto cestou poděkoval doc. Dr. Ing. Václavu Vostrovskému, Ph.D. za vedení této práce. Dále bych rád poděkoval zaměstnancům firmy Linde za možnou inspiraci na jejich řešení datové základny a následné spolupráce.

Navrhování datové základny databázových evidencí

Projecting a data base of database records

Souhrn

Práce se zabývá navrhováním datové základny. K této tematicce v první (teoretické) části objasní autor základní pojmy z této tematiky. Následně je v práci navržen postup pro návrh těchto základen v jednotlivých krocích, které nejsou vázány na konkrétní implementaci, ale popisují obecný univerzální postup pro návrh základny databázových evidencí.

V praktické části je tento postup aplikován na příklad z praxe. Samotný postup není tvořen jen vlastním datovým modelováním, ale jeho součástí je i stanovení cílů, vypracování analýzy, dokumentace a předpovědi budoucího vývoje návrhu. Analýza je aplikována na strukturu databáze reálné firmy a na základě jejích výsledků je řešen návrh nového východiska. Vlastní návrh uvedený v praktické části práce slouží primárně k demonstrování navrženého postupu.

Summary

This work is focused on designing of data base. To this theme in the first (theoretical) part author illustrates the basic concepts of the subject matter. Then, in this work is designed procedure of designing data base in individual steps that are not tied to a specific implementation, but describe a general procedure for proposing universal base of database records

The practical part of this procedure is applied to a practical example. Procedure itself is not only formed their own data modeling, but it does include the setting of objectives, the development of analysis, documentation and prediction of the future of proposal. The analysis is applied to structure of the real company database and on results of the analysis is solved proposal of new basis. Own proposal made in the practical section is primarily used to demonstrate the proposed procedure.

Klíčová slova: Relačně databázová technologie, datová základna, datové modelování, datová normalizace, datová integrita

Keywords: Relational database technology, data base, data modeling, data normalization, data integrity

Obsah

1	Úvod	9
2	Cíl práce a metodika	10
3	Teoretická východiska	11
3.1	Základní pojmy	11
3.2	Modelování dat	14
3.2.1	3-úrovňový pohled	14
3.2.2	4-úrovňový pohled	15
3.3	Součásti relační databáze	15
3.3.1	Entita	15
3.3.2	Atribut	16
3.3.3	Primární klíč	16
3.3.4	Cizí klíč	16
3.4	Konceptuální modely	16
3.4.1	E-R model – Chenova notace	16
3.4.2	Binární E-R model	17
3.4.3	Notace UML	18
3.5	Relační modely	19
3.6	Návrh databáze	19
3.6.1	Definování cílů návrhu	19
3.6.2	Analýza požadavků	20
3.6.3	Definice entitních množin	22
3.6.4	Určení relací a kardinality mezi entitami	23
3.6.5	Definování atributů	26
3.6.6	Výběr primárních klíčů	26
3.6.7	Normalizace	27

3.6.8	Zajištění integrity	32
3.6.9	Dokumentace	33
3.6.10	Budoucí vývoj	34
4	Vlastní řešení	35
4.1	Definování cílů návrhu	35
4.2	Analýza požadavků	35
4.3	Definice entitních množin	37
4.4	Určení relací a kardinality mezi entitami	38
4.5	Definování atributů	39
4.6	Výběr primárních klíčů	39
4.7	Normalizace	42
4.8	Zajištění integrity	42
4.9	Dokumentace	45
4.10	Budoucí vývoj	47
5	Výsledky a diskuze	48
6	Seznam použitých zdrojů	49
6.1	Monografické zdroje	49
6.2	Internetové zdroje	49
7	Seznam použitých obrázků	50
8	Seznam použitých tabulek	51

1 Úvod

Lidé využívali databáze již od nepaměti, dříve ve formě archivování záznamů pomocí knih a šanonů. S nástupem vývoje informační technologie se tato služba přesunula na výkonnější médium, tedy na počítače (servery), hlavně kvůli stále vyšším požadavkům na zpracování dat ať již z hlediska objemu dat v databázi nebo z důvodu větší flexibility v oblasti zpracování dat. Tato bakalářská práce se bude zabývat výhradně relačními databázemi, avšak v této oblasti je nutno zmínit i ostatní technologie, které se dnes používají jako například objektové databáze nebo objektově relační databáze. Relační databázový model byl navržen na přelomu 60. a 70. let minulého století doktorem E. F. Coddem. Za dobu své existence je tato technologie dobře zdokumentovaná. Až v 90. letech minulého století přišla nová forma technologie a to objektové databáze, které vedly následně k vývoji objektově relační databáze jako kombinace předchozích dvou. Navzdory tomu, že relační databáze jsou z chronologického hlediska nejstarší formou těchto technologií, není pravdou, že jejich zastoupení mezi uživateli je nahrazeno využíváním novějších druhů této technologie. Spíše naopak se ukázalo, že relační databáze jsou pro některé typy úloh vhodnější než objektové databáze. Právě proto se práce bude zabývat tematikou relačních databází, které jsou v praxi pro většinu úkolů vhodnějším řešením. Navzdory tomu, že relační databáze existují už dlouho, jsou stále v mnoha případech špatně navrženy. Práce proto objasní základní terminologii a postup návrhu databáze.

2 Cíl práce a metodika

Cíl práce

Bakalářská práce je tematicky zaměřená na problematiku navrhování datových základů relačních databázových evidencí. Cílem práce je:

- a) Vymežit teoretické principy problematiky relačních databází, navrhování základů včetně jejich datové normalizace
- b) zmapovat současnou situaci v této záležitosti, identifikovat požadavky kladené na takovéto evidence se zřetelem na eliminaci případných datových redundancí
- c) navrhnout a následně ověřit možnost řešení těchto požadavků na konkrétním příkladu z praxe
- d) ověřené záležitosti zobecnit pro další použití

Metodika

Použitá metodika zadané bakalářské práce bude založena na studiu a analýze dostupných informačních zdrojů a existujících řešení v dané oblasti. Stěžejní pro vypracování této závěrečné práce budou metody a techniky relačně databázové technologie v kontextu s modelováním datové základny. 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.

3 Teoretická východiska

Tato kapitola jasně definuje základní pojmy z oblasti databázových technologií, které budou zásadní při vytvoření obecného návrhu modelování databáze a jeho následného ověření na konkrétním modelu.

3.1 Základní pojmy

Databáze

Databáze je utříděný soubor informací (dat), který není závislý na aplikačním programu a je vytvořen za účelem možnosti vkládání nových záznamů, úpravy (aktualizace) a případného mazání záznamů.

Databázový server

„Pojem představuje soubor programových prostředků určených pro práci s daty, včetně organizace a realizace přístupu klientů k těmto datům.“ [2, s.13]

Schéma databáze

„Je popis dat na úrovni srozumitelné uživateli databáze a odpovídající používané vrstvě software. Běžně se pod schéma databáze zahrnují i IO.“ [3, s.7]

Ukládání dat v databázích

Je potřeba rozlišit fyzickou a logickou práci s daty potažmo jejich ukládání. Běžný uživatel se k vlastnímu fyzickému ukládání dat vůbec nedostane a pracuje s daty jen na logické úrovni. Samotné fyzické uložení dat obstarává SŘBD (viz odstavec Systém řízení báze dat). Vlastní logické ukládání probíhá ukládáním do dvourozměrných tabulek, jejíž řádky se nazývají záznamy a sloupce atributy.

Relační databázový model

Relační model byl definován Dr. Edgarem F. Coddem v roce 1970 v jeho práci nazvané Relační model dat pro velké sdílené databanky jako reakce na nevyhovující vlastnosti doposud používaných databází jako nízká integrita, redundance dat a hlavně přílišná závislost dat na jejich fyzickém uložení. Jakožto matematik se rozhodl tyto problémy vyřešit pomocí matematických metod a struktur. Odsud vychází název relační databázový model, jelikož je vytvořen ze dvou matematických oblastí, a tedy teorie množin (její součástí jsou relace) a predikátová logika prvního řádu. Název tedy není odvozen od vztahů (relací) mezi tabulkami, jak je často špatně vykládáno.

„Relační databáze ukládá data ve vztazích, které uživatel vidí jako tabulky. Každý vztah je složen z uspořádaných n-tic, neboli záznamů, a atributů, neboli polí.“ [1, s.47]

Databázové tabulky v relační databázi

Podmínky relačnosti definované Dr. E. F. Coddem v šedesátých až sedmdesátých letech minulého století:

- *„Všechna data v databázi jsou uložena v tabulkách.*
- *Fyzická struktura dat a jejich uložení jsou nezávislé a od uživatele odstíněné (tedy neexistují pro uživatele viditelné přístupové cesty – včetně indexů).*
- *Předpokládá se existence databázového jazyka, který umožňuje realizovat minimálně operace selekce, restrikce, projekce a spojení.“ [2, s.14-15]*

Dr. E. F. Codd dále definuje 12 základních pravidel pro relační databázové systémy:

- *„Informační pravidlo:
Všechny informace v relační databázi jsou vyjádřeny explicitně na logické úrovni jediným způsobem - hodnotami v tabulkách.*
- *Pravidlo jistoty:
Všechna data v relační databázi jsou zaručeně přístupná kombinací jména tabulky s hodnotami primárního klíče a jménem sloupce.*
- *Systematické zpracování nulových hodnot:
Nulové hodnoty jsou plně podporovány relačním SRŘBD pro reprezentaci informace, která není definována a to nezávisle na datovém typu.*

- *Dynamický on-line katalog založený na relačním modelu:*
Popis databáze je vyjádřen na logické úrovni stejným způsobem jako zákaznická data, takže autorizovaný uživatel může aplikovat stejný relační jazyk ke svému dotazu jako uživatel při práci s daty.
- *Obsáhlý datový podjazyk:*
Relační systém může podporovat několik jazyků a různých módů použitých při provozu terminálu. Nicméně musí být nejméně jeden příkazový jazyk s dobře definovanou syntaxí, který obsáhle podporuje definici dat, definici pohledů, manipulaci s daty jak interaktivně, tak programem, integritní omezení, autorizovaný přístup k databázi, transakční příkazy apod.
- *Pravidlo vytvoření pohledů:*
Všechny pohledy, které jsou teoreticky možné, jsou také systémem vytvořitelné.
- *Schopnost vkládání, vytvoření a mazání:*
Schopnost zachování relačních pravidel u základních i odvozených relací je zachována nejen při pohledu na data, ale i při operacích průniku, přidání a mazání dat.
- *Fyzická datová nezávislost:*
Aplikační programy jsou nezávislé na fyzické datové struktuře.
- *Logická datová nezávislost:*
Aplikační programy jsou nezávislé na změnách v logické struktuře databázového souboru.
- *Integritní nezávislost:*
Integritní omezení se musí dát definovat prostředky relační databáze nebo jejím jazykem a musí být schopna uložení v katalogu a nikoliv v aplikačním programu.
- *Nezávislost distribuce:*
Relační SŘBD musí být schopny implementace na jiných počítačových architekturách.
- *Pravidlo přístupu do databáze:*
Jestliže má relační systém jazyk nízké úrovně, pak tato úroveň nemůže být použita k vytváření integritních omezení a je nutno vyjádřit se v relačním jazyce vyšší úrovně.“[5]

Systém řízení báze dat

Zprostředkovává komunikaci mezi databázovým systémem (potažmo informačním systémem) a daty v databázi. Je základním systémovým vybavením nad databází sloužícím k operacím s databází. Tento vztah je vyjádřen rovnicí:

$DB + \text{SRBD} = \text{DBS}$ (Databáze + Systém řízení báze dat = Databázový systém)

Během doby, kdy se databáze používají, se toto přesné označení poněkud zvolnilo a pod pojmem databáze tedy můžeme rozumět systém řízení báze dat i databázový systém. Ke správnému porozumění pojmu je třeba znát kontext použití pojmu.

Data v databázích

Je nutné si uvědomit, že velikost databáze se může pohybovat od jednotek MByte až k terabyte (TB) či petabyte (PB). Jelikož databáze může zabírat velikost až v jednotkách petabyte, je třeba se vyhnout veškerým duplicitám dat. Tento nedostatek ošetřuje operace zvaná normalizace dat (viz kapitola Normalizace). Data v databázi jsou nezávislá na aplikaci, o tuto funkci se stará programové vybavení.

3.2 Modelování dat

Tuto tematiku rozdělujeme na následující pohledy:

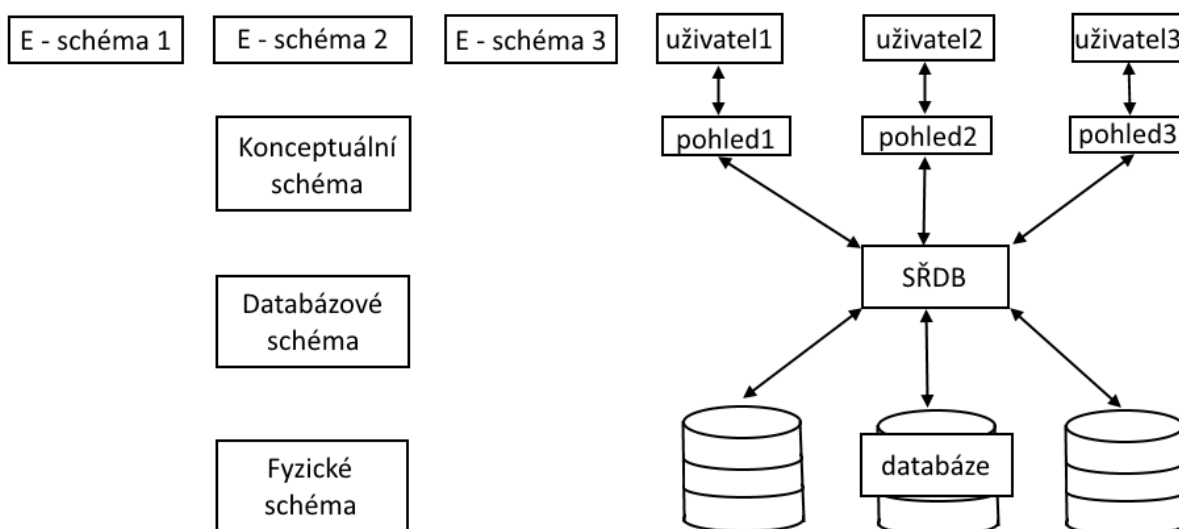
3.2.1 3-úrovňový pohled

Architektura tohoto přístupu k modelování dat vychází z předpokladu, že kategorie uživatelů mají různý přístup k databázi skrze aplikace. Jednotlivé externí schémata, popsána různými konceptuálními modely, definují rozdílné zobrazení atributů pro stejný typ entitní množiny různým kategoriím uživatelů.

„Konceptuální schéma představuje unifikovaný konceptuální pohled na objekty a jejich vztahy ze všech aplikací (v rámci budovaného IS) a slouží nejen k popisu toho, „o čem“ IS je, ale také jako jeden ze zdrojových dokumentů pro implementaci IS v prostředí příslušného SRBD.“ [3, s.17]

3.2.2 4-úrovňový pohled

V současné době se využívaná architektura. Na obrázku (viz Obrázek 1) vlevo je zobrazen teoretický model této architektury. V pravé části obrázku (viz Obrázek 1) se nachází vyobrazení využití této architektury v reálném prostředí. Uživatel (skupina uživatelů) má zobrazenou jen část databáze skrz pohledy, vytvořených na základě externích schémat. Uživatel při přístupu do databáze vznáší požadavky na systém řízení báze dat a ten vytvoří dotaz na základě podkladu od uživatele (zjednodušení složitých dotazů uživateli na jednodušší) a až poté vrátí systém řízení báze dat výsledek uživateli. Tímto je zajištěna nezávislost fyzických dat na aplikaci a opačně. Aplikace pracuje jen s logickou úrovní dat. Koncový uživatel do ukládání fyzických dat vůbec nezasahuje. Tuto prioritu má pouze správce dat pro možnost optimalizace. [3, s.17]



Obrázek 1 - Hierarchie abstrakcí a jejich realizace (zdroj: POKORNÝ, J.: Databázové systémy. ČVUT Praha 2013. s. 18)

3.3 Součásti relační databáze

Každá relační databáze se skládá minimálně z těchto částí, kterými jsou entita, atribut, primární klíč a cizí klíč

3.3.1 Entita

Je objekt, který zachycuje část reálného světa a je zanesen do datového modelu. Každá entita se skládá minimálně z jednoho atributu. V databázích se entity zobrazují jako řádky tabulky. Celou tabulku označujeme entitní množina.

3.3.2 Atribut

Vyjadřuje vlastnost dané entity, pro kterou je vytvořen. Data v atributu označujeme jako hodnoty atributu. Hodnoty atributů lze v databázích omezit na určité datové typy a následně i tyto datové typy omezit na požadované hodnoty (například omezení čísla pomocí požadovaného intervalu). Každý atribut musí být atomický pro další práci s daty (tuto podmínku zajišťuje proces normalizace).

3.3.3 Primární klíč

V každé tabulce musí být definován primární klíč, tedy jednoznačný identifikátor, který je pro každý záznam unikátní. V některých případech je jednosloupcový primární klíč nevyhovující, a proto lze zvolit složený primární klíč, který obsahuje n atributů spojených do primárního klíče (každý záznam je unikátní právě kombinací n atributů definovaných jako primární klíč).

3.3.4 Cizí klíč

„Je sloupec či skupina sloupců, které jsou propojeny na primární klíč v jiné tabulce.“ [2, s.16]

Vytváří vztah mezi dvěma tabulkami. Slouží k zachování integrity při operacích (např. mazání).

3.4 Konceptuální modely

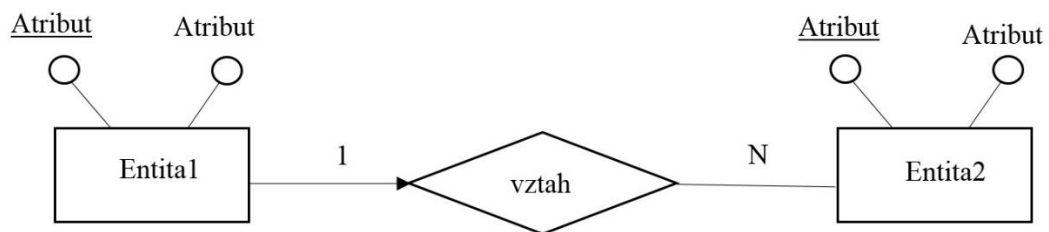
Modely se v relačních databázích skládají z entit, atributů a vztahů. Při vytváření modelu se klade velký důraz na jeho reálný základ, napříč tomu se při vytváření modelu musí reálný základ značně zobecnit na důležité údaje, se kterými potřebujeme dále pracovat. Jako první zkonstruoval a zdokumentoval takovýto model P. Chen v roce 1976, a hovoříme o E-R modelu (Entity-relationship model). Tento typ modelu prochází stále vývojem, a proto dnes existuje již celá rodina E-R modelů.

3.4.1 E-R model – Chenova notace

„Na entitu se v modelu kladou dva požadavky: má nezávislou existenci a je jednoznačně odlišitelná od ostatních entit.“ [3, s.19]

Toto téma souvisí s normalizací dat a primárními klíči. (viz kapitoly Normalizace a Primární klíč)

Jde o konvenci pro označování schématu při návrhu modelu. Entity jsou ve schématu představovány obdélníky, jejich atributy kolečky a vztah mezi jednotlivými entitami je zobrazován kosočtvercem, kdy se dále specifikuje směr závislosti pomocí šipky a kardinalita závislosti 1:1, 1:N, M:N. Problém nastává ve chvíli, kdy za předpokladu, že entita1 může být přiřazena k N entitám2, ale neznamená to, že entita2 může být N-krát přiřazena k entitě1. Kardinalita se vždy čte ze schématu zleva doprava a pro odstranění této chyby se obrácená kardinalita jednoduše nepíše. Dále se v schématu vyznačuje umístění primárního klíče pomocí podtržení názvu daného atributu (viz Obrázek 2).

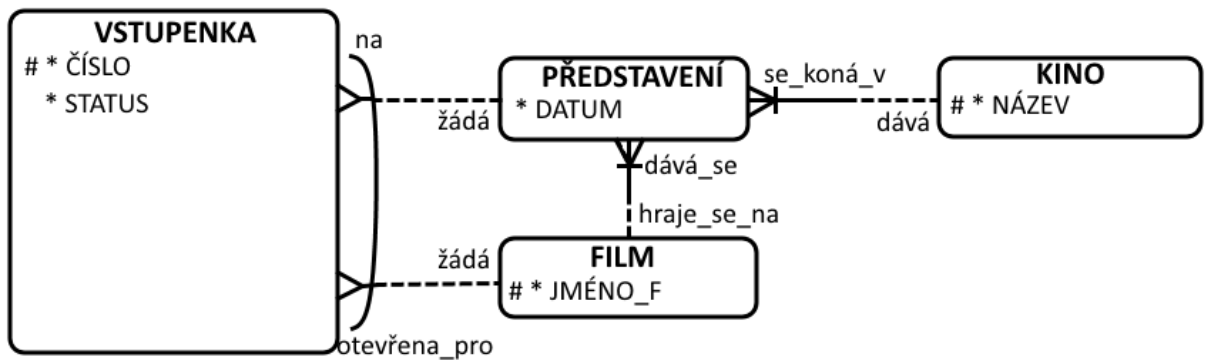


Obrázek 2 – Chenova notace, relace 1:N

3.4.2 Binární E-R model

Tento typ modelu rozlišuje kardinalitu pouze 1:1 a 1:N. Primární klíč je označen znakem #. Kardinalita je ve schématu zobrazena pomocí takzvané „vraní nohy“ na konci (viz Obrázek 3). Dále se pomocí plné čáry zobrazuje povinný vztah entit a pomocí přerušované čáry nepovinný.

„Binární E-R model vyžaduje určitý styl myšlení, který je odlišný od obecného E-R modelu. Pro někoho může být neexistence možnosti typu vztahu s kardinalitou M:N problémem. V tom případě je nutná dekompozice takového typu vztahu do dvou typů vztahů s kardinalitami N:1.“ [3, s.27] (viz Obrázek 3).



Obrázek 3 - Vylučné vztahy v binárním modelu (zdroj: POKORNÝ, J.: Databázové systémy. ČVUT Praha 2013. s. 26)

3.4.3 Notace UML

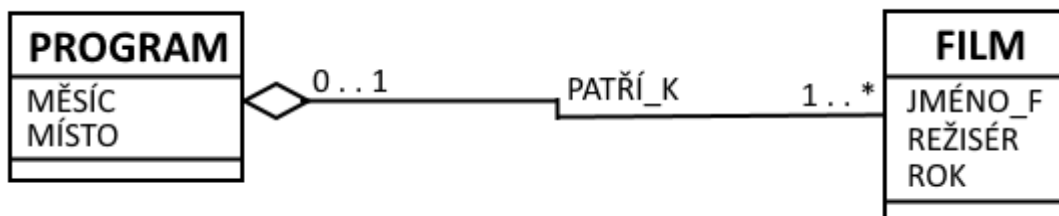
Jazyk UML slouží obecně pro navrhování struktury programu. Obsahuje i část jazyka, která slouží pro navrhování schématu tříd z E-R modelu. Při této konstrukci je třeba chápat třídy z UML diagramu jako typ entity z navrhování pomocí E-R diagramu. Při navrhování schématu se třídy skládají ze tří základních částí a těmi jsou:

„Popis třídy se provádí pomocí obdélníku rozděleného do tří částí:

- *jméno třídy*
- *atributy objektů třídy*
- *operace použité na objekty třídy*

Je zřejmé, že třetí část nemá v E-R modelu paralelu.“ [3, s.27]

Značení tohoto diagramu je následující: (viz Obrázek 4)



Obrázek 4 - Agregace UML (zdroj: POKORNÝ, J.: Databázové systémy. ČVUT Praha 2013. s. 28)

3.5 Relační modely

Tyto modely využívají databázové schéma, kde se schéma zachycuje pomocí tabulek. Jednotlivé sloupce reprezentují atributy a jednotlivé řádky reprezentují entity (respektive instance každé entity). Dále se tabulka skládá ze záznamů, což je jeden atribut vybrané entity. Pojem relační modely vychází z podobnosti mezi relačními modely a matematickou strukturou zvanou relace.

„Relační databáze je potom dána množinou relací.“ [3, s.27]

3.6 Návrh databáze

Pro všechny kroky návrhu platí, že je nesmírně důležité být po celou dobu návrhu v kontaktu se zadavatelem a uživateli. Informace, které takto získáme, jsou stěžejní pro návrh. Je také nesmírně důležité porozumět všem detailům a omezením, které se návrhu týkají.

Vlastní návrh probíhá od fáze definice entitních množin až po fázi zajištění integrity. Pro tyto fáze je doporučeno používat E-R diagramy pro jednoduchý a přehledný popis návrhu.

3.6.1 Definování cílů návrhu

Na počátku procesu návrhu databáze je třeba si určit cíle. Tyto cíle budou určovat, jaké požadavky má model splňovat po jeho dokončení. Cíle říkají, co má být splněno, neříkají jak. V první fázi je potřeba si uvědomit, jaké kritéria naše databáze musí splňovat na obecnější úrovni. Tyto cíle se částečně určí na základě rozhovoru se zadavatelem o tom, co ho vedlo k potřebě vytvořit nový systém. Tímto částečně zjistíme nedostatky, kterým se musíme vyvarovat.

Je rozlišeno 5 základních doporučení, kterých by se správná databáze měla držet a čím více se podaří při návrhu databáze obsáhnout těchto doporučení, tím lépe je databáze připravena pro všechny její funkce.

- *„Databáze podporuje plánované i neplánované získávání informací – Databáze musí uchovávat data nutná pro plnění informačních požadavků známých v době návrhu databáze, ale i požadavků, které budou zadávány uživatelem.“*

- **Tabulky jsou konstruovány správně a efektivně** – Každá tabulka v databázi představuje jednu entitu, je složena z relativně odlišných složek, udržuje na minimu redundanci dat a v databázi je identifikována polem s unikátní hodnotou.
- **Integrita dat je vynucena na úrovni složek, tabulek a vztahů** – Tyto úrovně integrity pomáhají garantovat, že datové struktury a jejich hodnoty budou v libovolném okamžiku platné a přesné.
- **Databáze podporuje business pravidla daná konkrétní organizací** – Data musí poskytovat platné a přesné informace, které mají význam pro podnikání
- **Databáze umožňuje budoucí rozšíření** – Databázová struktura by měla být snadno modifikovatelná, nebo rozšiřitelná podle potřeb dané organizace.“ [1, s.58]

Výhody dobrého návrhu

Za předpokladu, že si nejprve nastavíme výše uvedené cíle při tvorbě návrhu databáze, následně můžeme velmi těžít z tohoto postupu. Na první pohled se to nemusí zdát, ale čas, který vynaložíme na definování správných cílů a analýzy konkrétního návrhu, se vrátí při další práci s databází. Hlavními výhodami jsou:

- „Databázová struktura je snadno udržovatelná a modifikovatelná
- Data jsou snadno modifikovatelná
- Informace se dají snadno získat
- Aplikace pro koncového uživatele se dají snadno navrhnout a implementovat“ [1, s.59]

3.6.2 Analýza požadavků

Před zahájením návrhu je třeba analyzovat doposud používané řešení. Tato fáze analyzuje nedostatky současné struktury a zjistí požadavky na budoucí vývoj. V této fázi je nutné analyzovat požadavky zadavatele, pro kterého je dané řešení připravované. Jde o analýzu potřeb zadavatele, tedy vymezení oblasti, kterou bude databáze obsahovat. Dále se jedná o komunikaci se zákazníky, buď formou rozhovoru, nebo dotazníku. Jedná se tedy hlavně o sběr dat, kterými je nutné se řídit při návrhu. Mimo data, která se týkají aktuálního stavu

podniku, je třeba získat i představu vedení o možném růstu podniku, který by mohl přímo ovlivnit požadavky na databázi v budoucím vývoji (hlavně v oblasti struktury databáze).

Získávání dat

Při tomto kroku je důležité zjistit všechna média, na která podnik doposud své záznamy prováděl. Může se jednat o databázový systém, tabulkový procesor nebo dokonce papírovou podobu. Informace ze všech těchto médií je třeba shromáždit a zjistit souvislosti ve struktuře, kterou analyzovaný podnik pro zachycování dat využívá. V tuto chvíli je důležitý podrobný rozhovor s uživateli, kvůli přesnému porozumění všech souvislostí. Tato fáze se nesmí vynechat, protože na ní závisí veškeré další kroky a nedůslednost v této fázi se stává kritickou pro další postup.

Následným krokem je důkladné informování uživatelů. Všechny způsoby ukládání dat z oblasti, kterou se databáze bude zabývat, je třeba sloučit do jednoho (toto zajistí následný návrh) a uživatelé musí být dobře informováni o nutnosti udržování aktuálních informací v tomto systému, jinak dochází k nekonzistentnosti dat.

Rozhovory s managementem

Tento krok slouží k předběžné představě o zaměření návrhu. Hlavní fází této etapy je zaměření se na budoucí růst, který by ovlivnil návrh. Je třeba promyslet podobu návrhu tak, aby při následném doplnění nebyl problém se strukturou.

Rozhovory s uživateli

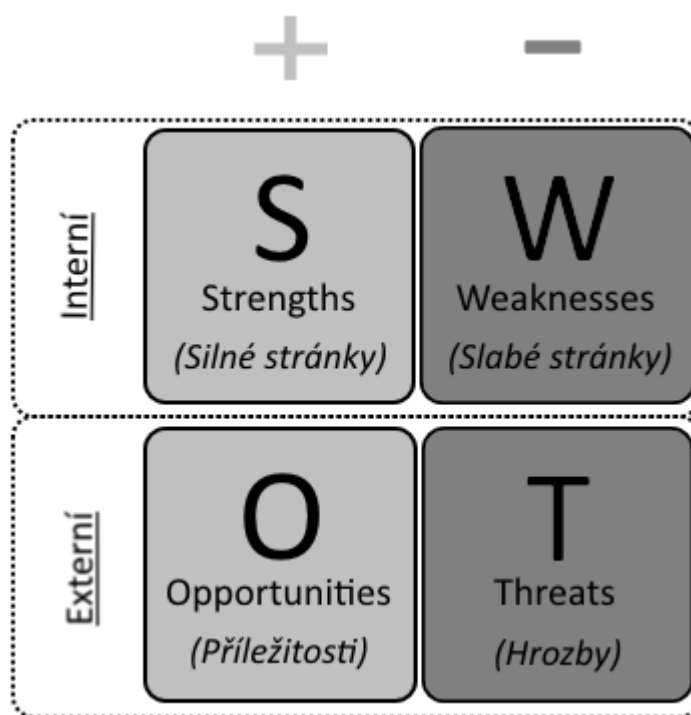
Záměrně tato fáze přichází až po rozhovorech s managementem, kde jsme získali obecnou představu o budoucím návrhu. Nyní je potřeba získat detailní informace od uživatelů, kteří budou navrhovanou databázi využívat. Je tedy nutné zjistit od uživatelů, jak s dosavadním systémem pro evidenci pracovali, jaká data jsou důležitá, případně jaká mají tyto data charakter. Tato fáze je velmi zásadní pro vlastní návrh. Zjistíme tak, která data je potřeba v databázi uchovávat, a která nikoliv.

Dále je potřeba zjistit, s čím byli uživatelé nespokojeni na dosavadním řešení, abychom se těchto nedostatků v návrhu vyvarovali.

Vyhodnocení požadavků

K vyhodnocení shromážděných poznatků o chování a potřebách použít nástroj pro analýzu. Jedním z kvalitních prostředků je využití SWOT analýzy. SWOT analýza rozděluje vlivy

na interní a externí. Interní určují vnitřní fungování systému a proto je jednodušší jejich změna při vyhodnocení. Externí určují s jakými vlivy je potřeba počítat při dalším vývoji. Dále zde platí rozdělení na kladné a záporné vlivy. Mezi kladné patří silné stránky a příležitosti a mezi záporné slabé stránky a hrozby (viz Obrázek 5). Tento proces jednoznačně vymezuje vlastnosti, na které je potřeba se při návrhu primárně zaměřit.



Obrázek 5 - Grafické znázornění SWOT analýzy

3.6.3 Definice entitních množin

Na základě analýzy požadavků jsme získali představu o obsahu databáze. Nyní začíná vlastní proces návrhu. Jednotlivé kroky jsou seřazeny v pořadí, ve kterém se provádí. Ideální nástroj pro zachycení tohoto procesu je E-R diagram, který přehledně znázorní většinu kroků návrhu.

Předběžný seznam entitních množin

Prvním krokem je ze získaných dat vytvořit předběžný seznam stěžejních vlastností (atributů), které budeme v databázi uchovávat a záznamy tohoto seznamu rozřadit do skupin, které spolu souvisí.

Pojmenování tabulek

Následující pravidla pomáhají udržet názvy entitních množin jednoznačné, popisné a smysluplné:

- Název je podstatné jméno
- Název se nesmí opakovat (musí být pro každou tabulku jedinečný)
- Pojmenování musí být srozumitelné všem, uživatelům pracujícím se systémem
- Jméno musí jednoznačně identifikovat entitu, kterou zaznamenává
- Co nejmenší počet slov v názvu
- Pro název nepoužívat zkratky a akronymy [1, s.158]

3.6.4 Určení relací a kardinality mezi entitami

Relace znázorňuje vztah mezi dvěma tabulkami, které spolu logicky souvisí. Vztah je zachycen v E-R diagramu jako kosočtverec se slovním názvem uvnitř. Název představuje vzájemný vztah dvou entitních množin a je většinou popsán slovesem. Kardinalita vztahu značí četnost entity vůči druhé, které se vztah týká.

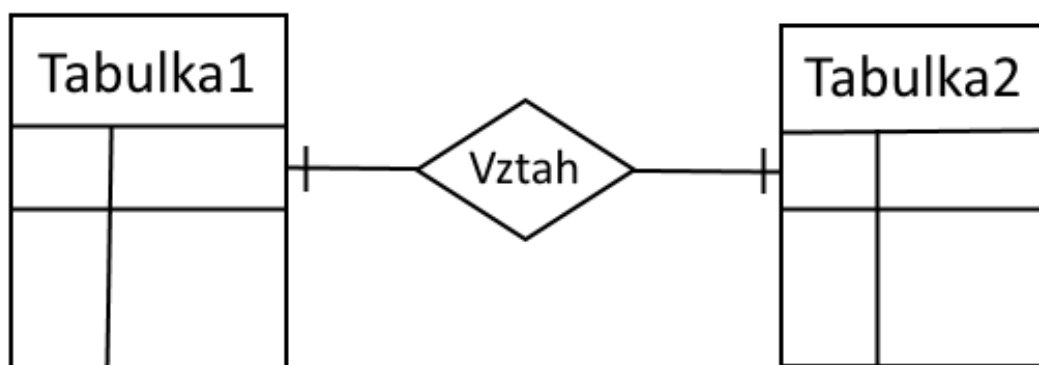
Existují čtyři základní typy vztahů:

Relace jedna ku jedné (1:1)

Jedná se o základní typ vztahu. Každý záznam v první tabulce odpovídá právě jednomu záznamu v druhé tabulce a zároveň každý záznam v druhé tabulce odpovídá právě jednomu záznamu v první tabulce (viz Obrázek 6).

Praktický příklad:

Každá universita má jednoho rektora. / Každý rektor působí na jedné universitě.



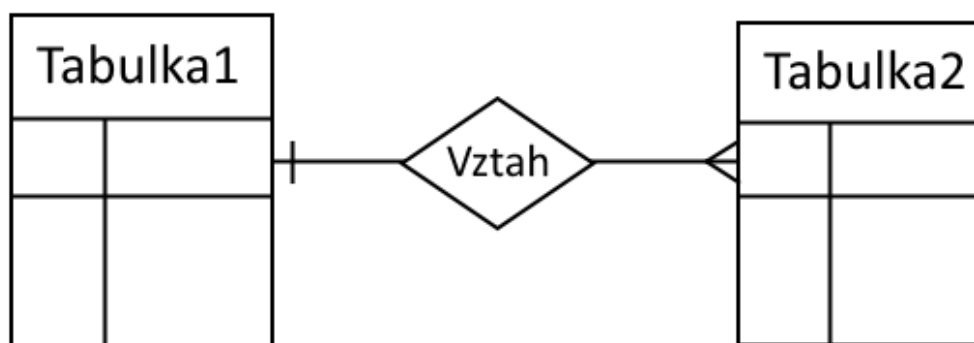
Obrázek 6 - Vztah 1:1 (zdroj: LACKO, L.: ORACLE. Správa, programování a použití databázového systému. Computer Press Brno 2010. s. 17)

Relace jedna ku více (1:N)

U tohoto typu relace platí, že každý záznam v první tabulce je možné propojit s více záznamy z tabulky druhé a zároveň každý záznam z druhé tabulky lze propojit s jedním záznamem z první. (viz Obrázek 7)

Praktický příklad:

Každá universita vzdělává více studentů. / Každý student je vzděláván na jedné universitě.



Obrázek 7 - Vztah 1:N (zdroj: LACKO, L.: ORACLE. Správa, programování a použití databázového systému. Computer Press Brno 2010. s. 17)

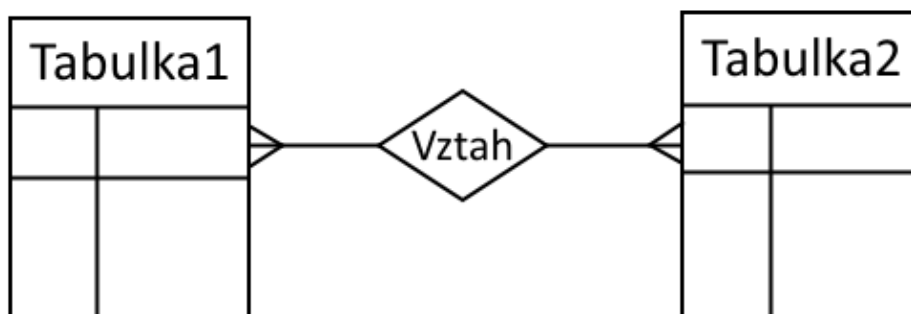
Relace více ku více (N:M)

V tomto vztahu relace platí, že každý záznam z první tabulky je možné propojit s více záznamy z druhé tabulky a zároveň každý záznam z druhé tabulky je možné propojit s více záznamy tabulky první (viz Obrázek 8). V databázi tento vztah není podporován, a proto se

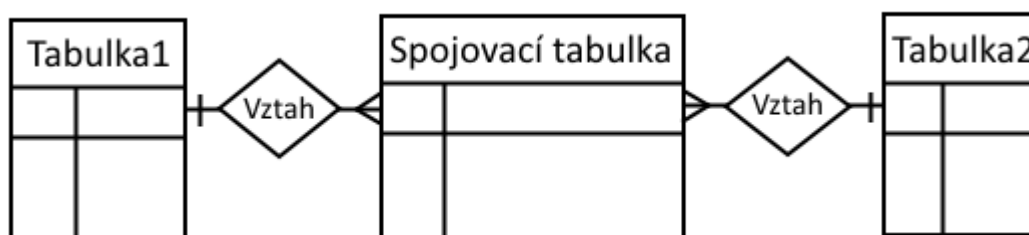
tento typ relace řeší pomocí takzvané spojovací (vazební) tabulky (viz Obrázek 9), která slouží jako propojení původních dvou tabulek. Mezi vazební a původními tabulkami vždy platí vztah 1:N.

Praktický příklad:

Každá universita uděluje více titulů. / Každý titul je udělován více universitami.



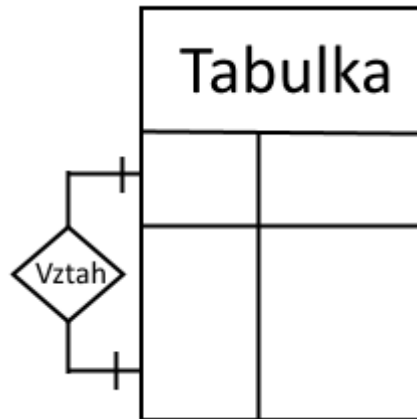
Obrázek 8 - Vztah N:M (zdroj: LACKO, L.: ORACLE. Správa, programování a použití databázového systému. Computer Press Brno 2010. s. 18)



Obrázek 9 - Vztah N:M, rozložení pomocí vazební tabulky (zdroj: LACKO, L.: ORACLE. Správa, programování a použití databázového systému. Computer Press Brno 2010. s. 18)

Unární relace

Toto je speciální typ relace, pro kterou platí, že vazba je aplikovaná na jedné tabulce mezi jejími záznamy. Tento typ relace se v praxi využívá pro zobrazení hierarchického vztahu nadřizený - podřizený. (viz Obrázek 10)



Obrázek 10 - Unární relace (zdroj: LACKO, L.: ORACLE. Správa, programování a použití databázového systému. Computer Press Brno 2010. s. 18)

3.6.5 Definování atributů

Krok definice entitních množin definoval jednotlivé entitní množiny na základě seskupení stěžejních atributů. V tomto kroku budou doplněny všechny entitní množiny o atributy, které se dané entitní množiny týkají. Atributy jsou nyní dále nedekomponovatelné (pokud je stále možný další rozklad, je tento atribut převeden na entitu, která je dále reprezentována atributy dekomponovaného atributu a určí se nově vzniklé relace). Všechny atributy musí být přímo závislé na primárním klíči dané relace (tímto krokem se automaticky splní 2. normální forma).

3.6.6 Výběr primárních klíčů

Prvním krokem je u každé tabulky určit alespoň jeden kandidátní klíč. Kandidátní klíč musí splňovat tyto požadavky:

- Nesmí být vícesložkové pole (např. Bydliště – Vodičkova15 Praha)
- Musí mít jedinečné hodnoty
- Nesmí obsahovat hodnoty NULL (nepřítomnost hodnoty)
- Může být složený z více atributů
- Jeho hodnota se mění jen výjimečně [1, s.184]

Pro jednoznačnou identifikaci určíme u každé tabulky nejvhodnější z vybraných kandidátních klíčů, tento kandidátní klíč se stává primárním klíčem. Primární klíč se také často určí podle vztahu k dalším entitám, přičemž vztah je realizován prostřednictvím cizího

klíče. „Cizí klíč je atribut nebo skupina atributů identifikující jinou „nadřazenou“ entitu a jsou převzaty do „podřízené“ entity.“ [4, s.104]

3.6.7 Normalizace

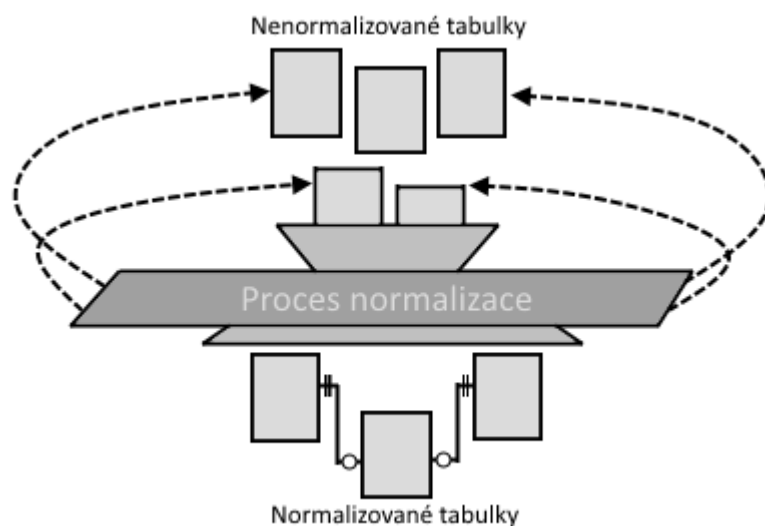
Jde o proces postupné aplikace normálních forem na tabulky. Normalizace slouží k odstranění redundancí v tabulkách a zvýšení efektivity. Samotný proces slouží ke kontrole strukturální správnosti dat a konzistenci datového modelu, ale neprověří reálný základ jednotlivých entit. Existuje 7 základních normálních forem. Obecně platí, že čím více normálních forem při tvorbě databáze respektujeme, tím je databáze přehlednější a lépe se s ní pracuje.

Ne vždy toto tvrzení musí platit, existují situace, kdy se od normalizace vyšších forem upouští, například za účelem zvyšování výkonu ve speciálních případech nebo při návrhu datových skladů. [2, s.19]

V takovém případě se v praxi končí zajištěním Boyce-Coddovy normální formy.

Při ověřování správnosti tabulky normálními formami vždy platí, že pokud chceme aplikovat na tabulku například 3NF, automaticky znamená, že musí zároveň platit všechny předešlé NF (tedy 2NF a 1NF) (viz Obrázek 11). Výjimkou je 0NF, která je protiklad 1NF.

U následujícího rozpisu jednotlivých normálních forem se pokusím jejich význam pro databázi vysvětlit na jednoduchých příkladech.



Obrázek 11 - Grafická reprezentace obecného normalizačního procesu, [1, str 61]

Nultá normální forma (0NF)

„Tabulka je v nulté normální formě tehdy, pokud se skládá alespoň z jednoho pole, které obsahuje více než jednu hodnotu, tedy není atomické.“ [2, s.19]

Příklad tabulky v 0NF:

Jméno	Učebna	Termín cvičení
Ekonomie	E220	Po 8:45
		St 12:15
Softwarové inženýrství	E260	Út 10:30
		St 12:15

Tabulka 1 - Nultá normální forma

První normální forma (1NF)

První normální forma platí, pokud všechny hodnoty v tabulce jsou již atomické, neboli všechny hodnoty v tabulce již není možno dále dělit.

Příklad:

Rozhodneme evidovat záznamy o zaměstnancích. Je tedy třeba evidovat atributy jméno, příjmení, bydliště. Problém nastává tehdy, když se do atributu bydliště rozhodneme zaznamenat například ulici a město zároveň.

Jméno	Příjmení	Bydliště
Jan	Novák	Ruská Praha
Pavel	Dvořák	Čínská Praha
Petr	Novák	Modrá Praha

Tabulka 2 - První normální forma před úpravou

Tedy ve chvíli, kdy na takto navrženou tabulku aplikujeme dotaz, který má jako podmínku výběru hodnotu atributu bydliště Praha, není tento dotaz možno provést, protože v atributu bydliště bude sice Praha, ale teoreticky se Praha může také nalézat v části, kde jsme chtěli evidovat ulici (teoreticky se ulice v jiném městě než Praha může také jmenovat Praha). Zjistíme tedy, že takto je tabulka zaměstnance navržena špatně a správně by měla být navržena, aby splňovala první normální formu, jméno, příjmení, město a ulice. Je tedy snaha o to, aby se v každé hodnotě atributu skládala právě z jednoho atributu.

Jméno	Příjmení	Město	Ulice
Jan	Novák	Praha	Ruská
Pavel	Dvořák	Praha	Čínská
Petr	Novák	Praha	Modrá

Tabulka 3 - První normální forma po úpravě

Druhá normální forma (2NF)

Samotná definice říká, že tabulka je ve druhé normální formě, pokud splňuje podmínky předchozích normálních forem a každý atribut, kromě atributu označeného jako primární klíč, je přímo závislý na primárním klíči. Tabulky, které nesplňují tuto normální formu, jsou klasickým případem redundance dat. Tato normální forma se týká pouze tabulek se složeným primárním klíčem, pro tabulky s jedním primárním klíčem je druhá normální forma automaticky zajištěna. Tento problém se řeší jednoduchým rozkladem na více tabulek.

Příklad:

Mějme tabulku zachycující atributy název firmy, název pobočky, telefon na pobočku a hlavní sídlo. Složený primární klíč je tvořen z atributů název firmy a název pobočky.

Název_firmy	Název_pobočky	Telefon_na_pobočku	Hlavní_sídlo
Alfa	Centrála	298 461 156	Praha
Alfa	Západ	289 466 148	Praha
Beta	Sever	245 846 354	Ostrava

Tabulka 4 - Druhá normální forma před úpravou

Z pojmenování atributů je zřejmé, že atribut telefon na pobočku není závislý na celém primárním klíči, ale jen na jeho části (atributu název pobočky). Tato situace nesmí nastat z důvodu nekonzistentnosti záznamů v rámci mazání. Ve chvíli, kdy z tabulky odstraníme všechny hodnoty atributu název_pobočky firmy Alfa, ztratíme i hlavní sídlo firmy. Tento problém se řeší rozkladem na dvě tabulky, z nichž jedna obsahuje atributy název firmy, id pobočky a hlavní sídlo a druhá id pobočky, název pobočky a telefon na pobočku.

Název_firmy	ID_pobočka	Hlavní_sídlo
Alfa	1	Praha
Alfa	2	Praha
Beta	3	Ostrava

Tabulka 5 - Druhá normální forma po úpravě 1

ID_pobočka	Název_pobočky	Telefon_na_pobočku
1	Centrála	298 461 156
2	Západ	289 466 148
3	Sever	245 846 354

Tabulka 6 - Druhá normální forma po úpravě 2

Třetí normální forma (3NF)

Definice říká, že relační tabulky splňují třetí normální formu, jestliže splňují 2NF a žádný atribut, který není primárním klíčem, není tranzitivně závislý na žádném klíči. Další možná definice zní: „*Tabulka je ve třetí normální formě tehdy, pokud je ve druhé normální formě a současně neexistují závislosti neklíčových sloupců tabulky.*“ [2, s.22]

Příklad:

Navrhujeme-li tabulku zaměstnanec, která zachycuje atributy id zaměstnance, jméno, příjmení, rok narození, funkce a plat.

ID_zaměstnanec	Jméno	Příjmení	Rok_narození	Funkce	Plat
1	Jan	Novák	1960	technik	20000
2	Pavel	Dvořák	1962	vedoucí	30000
3	Petr	Novák	1968	technik	20000

Tabulka 7 - Třetí normální forma před úpravou

Tato tabulka není ani ve druhé normální formě, ale to ignorujeme, je totiž potřeba pochopit, co v tomto příkladu znamená tranzitivnost. Jde o nepřímou závislost atributu na jiném (tedy na primárním klíči, který v tomto příkladu bude id zaměstnance). Na první pohled je vidět, že atributy jméno, příjmení, rok narození a funkce přímo závisí na atributu id zaměstnance. Na druhou stranu je také vidět, že atribut plat není přímo funkčně závislý na atributu id zaměstnance, ale na atributu funkce. Tranzitivita znamená, že jestliže atribut funkce závisí

na atributu id zaměstnanec a zároveň atribut plat závisí na atributu funkce, znamená to, že atribut plat závisí na atributu id zaměstnanec (= tranzitivní závislost). Řešením tohoto problému je rozklad na dvě tabulky, z nichž jedna obsahuje id zaměstnanec, jméno, příjmení, rok narození, funkci a druhá funkci a plat.

ID_zaměstnanec	Jméno	Příjmení	Rok_narození	ID_funkce
1	Jan	Novák	1960	1
2	Pavel	Dvořák	1962	2
3	Petr	Novák	1968	1

Tabulka 8 - Třetí normální forma po úpravě 1

ID_funkce	Funkce	Plat
1	technik	20000
2	vedoucí	30000

Tabulka 9 - Třetí normální forma po úpravě 2

Ideálním řešením je u druhé tabulky přidat atribut id funkce a pomocí tohoto atributu navázat na první tabulku.

Boyce-Coddova normální forma

Definice zní: „Relace R je v BCNF tehdy a jen tehdy, když pro každou netriviální závislost $X \rightarrow Y$, kde X a Y jsou množiny atributů a zároveň Y není podmnožinou X , platí, že X je nadmnožinou nějakého klíče, nebo X je klíčem relace R . Jinak řečeno relace R je v BCNF tehdy a jen tehdy, když každý determinant funkční závislosti v relaci R je zároveň kandidátním klíčem relace R .“ [6]

Neboli všechna data jsou závislá na primárním klíči a ne mezi sebou.

Čtvrtá normální forma (4NF)

„Tabulka je ve čtvrté normální formě tehdy, pokud je ve třetí normální formě a popisuje jen jeden jediný fakt nebo souvislost.“ [2, s.22]

Pátá normální forma (5NF)

Tabulka je v páté normální formě, pokud se po přidání libovolného nového atributu rozpadne na více tabulek.

3.6.8 Zajištění integrity

Integrita vyjadřuje přesnost, platnost a konzistentnost dat v databázi. Jedná se o velmi důležitou vlastnost databáze, která přímo ovlivňuje budoucí komplexnost a přesnost dat, se kterými v databázi dále pracuje. Nedostatečná důslednost na dodržování integrity dat vytváří v databázích chyby, které dělají data nepřesná, nebo dokonce neplatná. Existují čtyři typy integrity, na které je potřeba se při návrhu zaměřit a předejít tak chybám.

Integrita na úrovni tabulky (integrita entity)

V rámci tabulky nevznikají žádné duplicity v rámci záznamů a jednotlivé atributy záznamu, které definují vlastnosti záznamů, jsou jedinečné, jednoznačně definované a hodnoty atributu, které identifikují jednotlivý záznam (primární klíč) neobsahují hodnotu NULL.

Integrita na úrovni pole (doménová integrita)

Hodnoty každého atributu jsou konzistentní, platné a přesné. Každý atribut v modelu musí být zařazen do domény. V rámci jedné databáze jsou všechny atributy, označující stejný typ vlastnosti, stejně definované (např. jestliže je atribut cena v jedné tabulce definován jako desetinné číslo, musí být všechny atributy cena v rámci databáze definovány jako desetinné číslo). Doména definuje, jakých hodnot může atribut nabývat.

Doménová integrita určuje tyto charakteristiky:

- *„Datový typ (číslo, znak, datum)*
- *Délka (počet znaků, číslic)*
- *Rozsah (meze od-do)*
- *Přípustné hodnoty (dovolené vyjmenované výskyty)*
- *Formát (maska, určující strukturu hodnoty)*
- *Jedinečnost (kandidátní klíč)*

- *Přípustnost „NULL“ hodnoty*
- *Popis* “[4, s.107]

Integrita na úrovni vztahů (referenční integrita)

Zajišťuje spolehlivost mezi tabulkami v rámci relací. Data v tabulkách, která jsou na sebe navázána, jsou stále konzistentní při vkládání nových dat, úpravě dat nebo mazání dat.

Business pravidla

„Business pravidlo je formulace, která nějakým způsobem určuje omezení na určitou součást databáze, například na vlastnost atributu, nebo na charakteristiky některého vztahu.“ [1, s.274]

Jedná se o soubor pravidel, která jsou jedinečná pro každý návrh. Jde o interní požadavky zadavatele na jednotlivé vlastnosti atributů, například v rámci odvětví, ve kterém daná firma působí (např. v odvětvích, jako zemědělství, je možné uchovávat časové položky v rámci dnů, ale naopak u odvětví, jakým je podnikání na burze, je třeba zachycovat jednotlivé transakce s přesností na sekundy). Ovšem spíše než na odvětví, záleží na požadavcích jednotlivých zadavatelů. Z tohoto důvodu je velmi důležitá fáze analýzy.

Dále se jedná o pravidla logického významu (například datum dodání zboží nemůže být staršího data než datum objednání tohoto zboží). V rámci databáze se jedná o dva atributy, které na sobě nejsou závislé, ale právě business pravidla dávají logickou souvislost mezi těmito atributy, a proto zde aplikujeme výše uvedené omezení.

3.6.9 Dokumentace

Předposlední fází je vytvoření dokumentace k danému návrhu. Je třeba detailně zaznamenat celý postup návrhu a zdůvodnění jednotlivých kroků. Dokumentace zahrnuje i fázi analýzy, ve které je potřeba zhodnotit doposud používaný systém evidence a na základě rozhovorů vyhodnotit klady a zápory dosavadního řešení.

Dále zachycuje komplexní přehled o celém návrhu v jeho konečné podobě. Jednoznačně definuje všechny navržené tabulky a jejich význam. Definuje vztahy mezi tabulkami,

všechny atributy obsažené v tabulkách a jejich integritní omezení včetně business pravidel definovaných pro daný návrh. Pro každý atribut slovně určí jeho význam.

3.6.10 Budoucí vývoj

Tento krok slouží jako úvaha o dalším vývoji. Žádný návrh nedokáže vystihnout všechny budoucí požadavky zadavatele. Správný návrh se ovšem musí zaměřit na požadavky, které je možné v danou chvíli predikovat. K predikování požadavků slouží, mimo jiné, analýza požadavků. Je tedy důležité, aby návrh při doplnění požadavku nemusel být celý zrušen a navržen znovu. Návrh musí být navržen komplexně a v případě potřeby doplnění požadavku na jeho vlastnosti, vyřešit tuto situaci svou modifikací.

4 Vlastní řešení

V teoretické části byl autorem na základě získaných znalostí a informací vytvořen postup pro návrh databáze. Nyní v praktické části práce je tento postup demonstrován na konkrétním příkladu a ověřena tak možnosti tohoto řešení.

4.1 Definování cílů návrhu

Prvním krokem je definování cílů. Zadavatel je firma, která se zabývá obchodem s plyny, jako jsou technické plyny, plyny využívané ve zdravotnictví, chladiva, speciální plyny atd. Cíle jsou definovány v krátkých stručných větách. Po rozhovoru se zadavatelem jsou určeny cíle návrhu:

- Vytvořit návrh databáze připravený pro vlastní implementaci.
- Možnost pracovat s daty týkající se objednávek.
- Evidovat informace o dodavatelích a odběratelích.
- Evidovat komplexní informace o zásobách zboží.

4.2 Analýza požadavků

K práci s daty využívá zadavatel systém SAP HANA. Zadavatel hodnotí jako největší zápor dosavadního řešení nevyhovující strukturu z důvodu zastarání návrhu vůči aktuálním požadavkům. Jako nejdůležitější klad shledává rychlost systému. Tento klad vychází z konstrukce systému SAP HANA, ne ze struktury návrhu.

Analýza dosavadního systému

SAP HANA je informační systém, který využívá tato firma. Jedná se o databázový systém od společnosti SAP, založenou 1972 pěti bývalými zaměstnanci firmy IBM. Jedná se o databázovou platformu, která je určena pro velmi rychlé zpracování velkého objemu dat v reálném čase.

„Jednou z klíčových vlastností SAP HANA je umístění celé databáze pouze v operační paměti serveru (koncept „in-memory“). To znamená, že data se již z disků průběžně nenačítají a ani se na ně nezapisují, což zásadně zrychluje přístup k datům a veškeré operace s nimi. Další stěžejní vlastností SAP HANA je způsob ukládání informací. Na rozdíl od běžných SQL databází, se primárně ukládají „sloupcově“ (tzv. „column store“) a nikoli „řádkově“ (tzv.

„row store“), byť „řádkové“ ukládání SAP HANA také podporuje. Například v databázi obsahující data o všech obyvatelích České republiky by byla spousta řádek osob s příjmením Novák. Ve „sloupcovém“ uložení bude příjmení Novák uloženo ve sloupci všech českých příjmení, a to pouze jednou.“ [7]

SWOT analýza

Na základě rozhovorů se zadavatelem, managementem a uživateli jsou shromážděna data, která budou pomocí SWOT analýzy rozřazena do jednotlivých skupin. Tyto skupiny znázorňují silné stránky, slabé stránky, příležitosti a hrozby.

Snahou je analyzovat strukturu databáze, kterou firma využívá a tím zjistit klady, které budoucí návrh nesmí postrádat, a zápory, kterých se budoucí návrh musí vyvarovat.

Silné stránky

Uvedení předností, dovedností, uvedení charakteristik, které jsou přínosné pro dosažení určitých cílů. Tato oblast se týká interních kladů struktury databáze.

1. Správně určené relace
2. Přesná dokumentace
3. Výkon

Slabé stránky

Uvedení položek, které mohou ohrozit dosažení daných cílů nebo vedou k úplné likvidaci. Tato oblast se týká interních záporů struktury databáze.

1. Dosavadní návrh nespĺňuje aktuální požadavky (struktura databáze)
2. Zbytečné atributy v tabulkách
3. Nejednoznačné pojmenování
4. V některých případech špatně zvolené primární klíče
5. Duplicity

Příležitosti

Jde o potenciální příležitosti, které firmě nebo systému mohou pomoci dosáhnout svých cílů. Tato oblast se týká externích kladů struktury databáze.

1. Využití analýzy aktuálního řešení pro nový návrh

Hrozby

Jedná se o oblast, která přináší potenciální rizika, jsou to tedy vnější podmínky, které mohou ohrozit dosažení daných cílů. Tato oblast se týká externích záporů struktury databáze.

1. Nekonzistentnost dat
2. Složitě úpravy aktuálního řešení

4.3 Definice entitních množin

Prvním krokem je rozřazení stěžejních atributů do skupin.

Předběžné stěžejní atributy pro navrhované řešení:

Číslo objednávky, datum uzavření objednávky, termín dodání, označení slevové skupiny, název zboží, cena, adresa, název dodavatele, telefon, název odběratele, kategorie zboží, kapacita skladu, název dopravce, cena za km, bezpečnostní skupina ADR

Nyní definujeme skupiny těchto atributů:

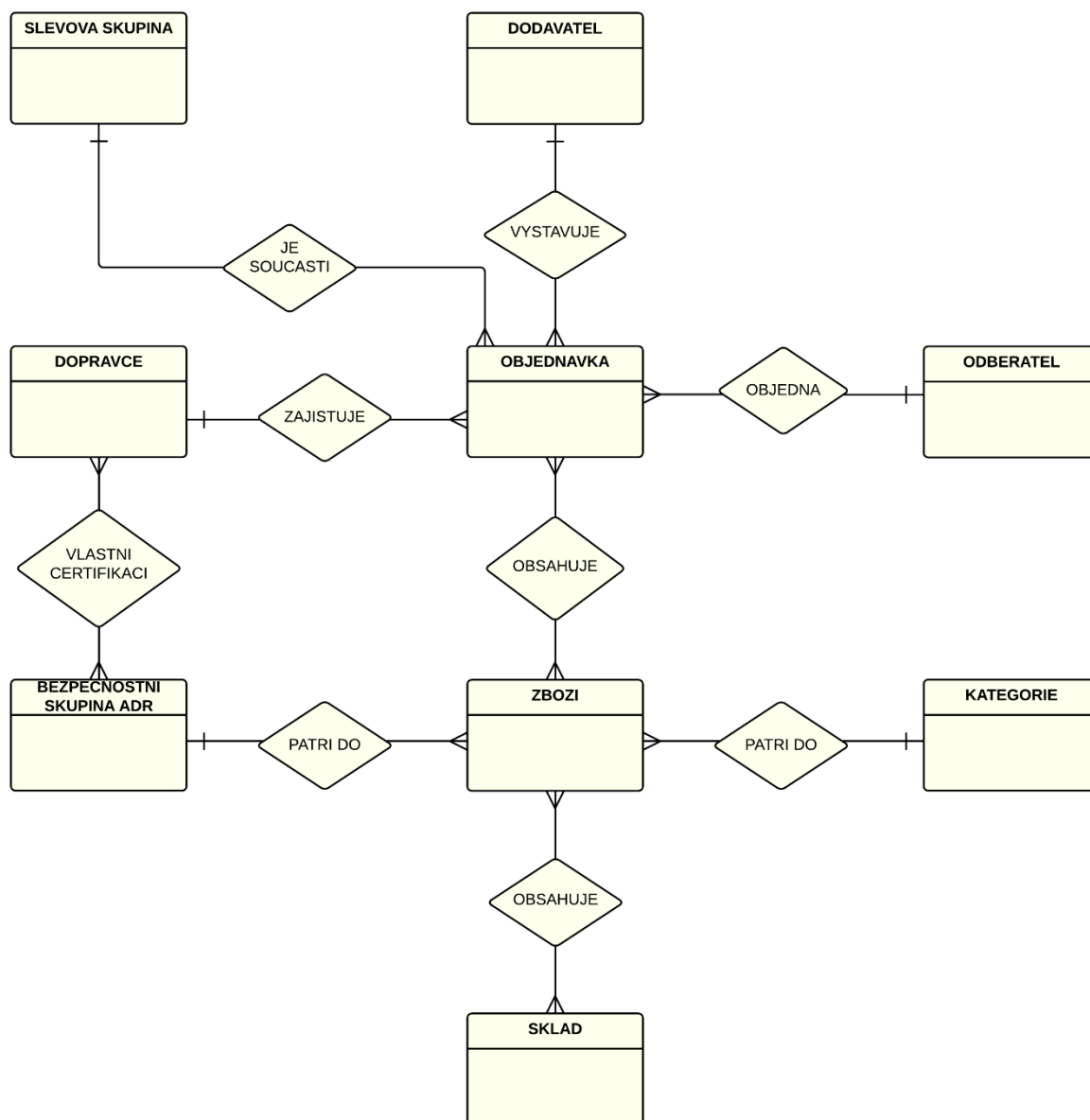
- Objednávka - číslo objednávky, datum uzavření objednávky, termín dodání
- Slevová skupina - označení slevové skupiny
- Zboží - název zboží, cena
- Dodavatel - název dodavatele, telefon, adresa
- Odběratel - název odběratele, telefon, adresa
- Kategorie zboží - kategorie zboží
- Sklad - kapacita skladu
- Dopravce - název dopravce, cena za km
- Bezpečnostní skupina ADR - bezpečnostní skupina ADR

Tímto jsou definovány skupiny, které znázorňují množinu tabulek, kterou bude databáze obsahovat. Jejich počet a struktura se mohou během procesu návrhu měnit. Názvy tabulek splňují pravidla definována v teoretické části zabývající se pojmenováním tabulek. Jedinou výjimkou je tabulka bezpečnostní skupina ADR, jejíž využití zkratky ADR je v tomto případě namísto kvůli českému ekvivalentu tohoto termínu (Evropská dohoda o mezinárodní silniční přepravě nebezpečných věcí).

4.4 Určení relací a kardinality mezi entitami

Nyní je potřeba mezi tabulkami vytvořit relace. Tato fáze opět vychází z informací získaných z analýzy dosavadní databáze, rozšířených o informace z rozhovorů s managementem a uživateli.

Na obrázku (Obrázek 12) je znázorněn E-R diagram zachycující definované vztahy.



Obrázek 12 - Praktická část, určení relací

V diagramu jsou použity dva typy vztahů (1:N a M:N) Každý vztah je vysvětlen na jednom příkladu (ostatní jsou interpretovány analogicky).

Relace 1:N (popis vztahu mezi tabulkami DOPRAVCE a OBJEDNAVKA):

Každý dopravce zajišťuje více objednávek. / Každá objednávka je zajištěna jedním dopravcem.

Relace M:N (popis vztahu mezi tabulkami OBJEDNAVKA a ZBOZI):

Každá objednávka obsahuje více zboží. / Každé zboží je obsaženo ve více objednávkách.

4.5 Definování atributů

Zatím jsou definované pouze stěžejní atributy pro návrh. V tomto kroku tabulky rozšíříme o další atributy.

- Objednávka - číslo objednávky, datum uzavření objednávky, termín dodání, množství, kontaktní osoba, doba splatnosti, měna
- Slevová skupina - označení slevové skupiny, id slevová skupina, násobitel, podmínka zařazení
- Zboží - název zboží, cena, inventární číslo, jednotky, stav, popis
- Dodavatel - název dodavatele, telefon, ulice, město, id dodavatel, popisné číslo, email, bankovní spojení, IČO, DIČ, právní osobnost
- Odběratel - název odběratele, telefon, ulice, město, id odběratel, popisné číslo, email, právní osobnost, odvětví podnikání
- Kategorie zboží – id kategorie, název, popis
- Sklad - kapacita skladu, ulice, město, id sklad, popisné číslo
- Dopravce - název dopravce, cena za km, id dopravce, telefon, email
- Bezpečnostní skupina ADR - id bezpečnostní skupina ADR, název, popis

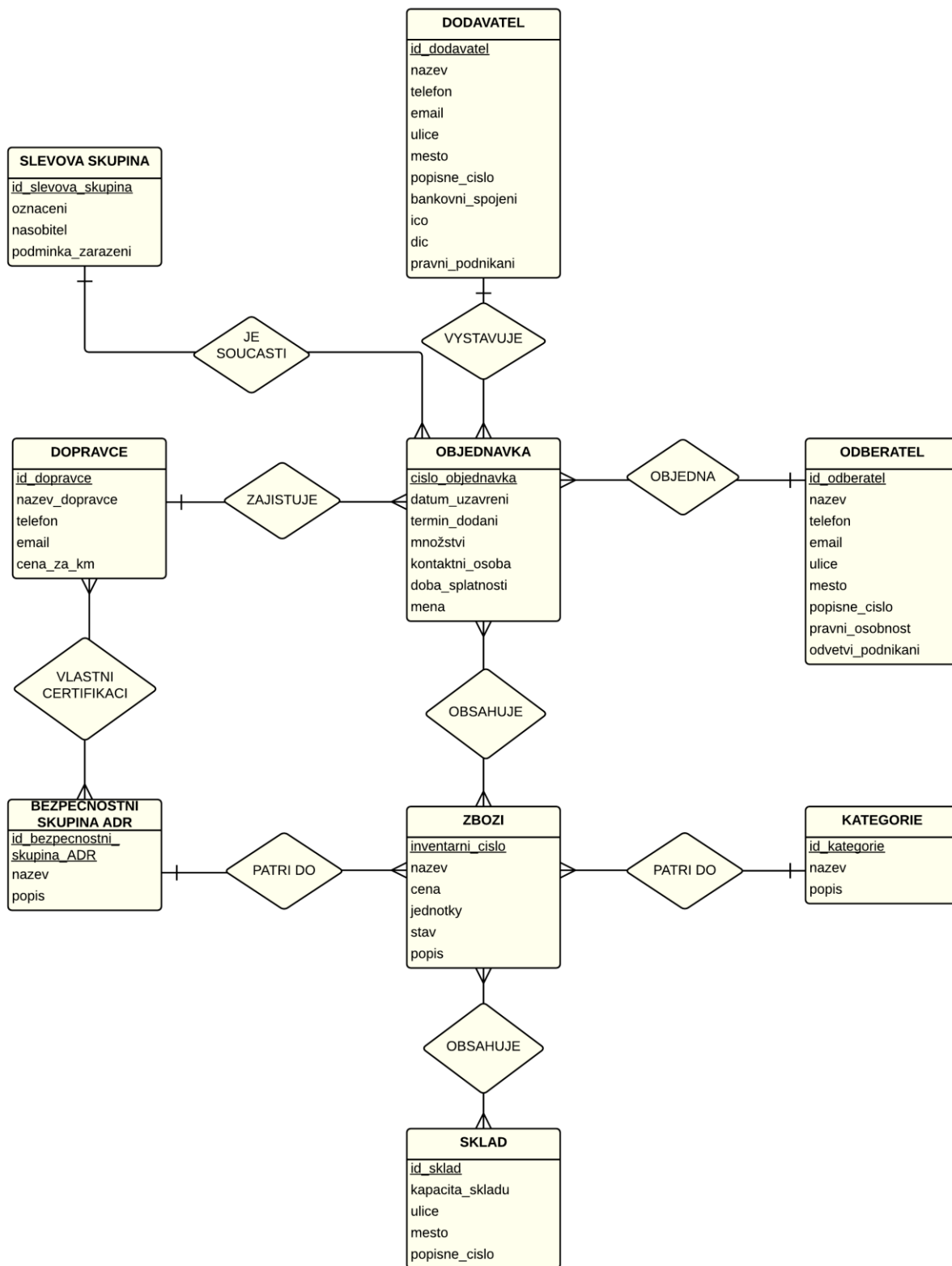
4.6 Výběr primárních klíčů

U každé tabulky se určí kandidátní klíče, které splňují požadavky definované v teoretické části věnující se výběru primárních klíčů. Hlavní důraz je kladen na jedinečnost hodnot a na ošetření hodnoty NULL (nesmí nastat). Nejvhodnější kandidátní klíč se stává primárním klíčem.

Příklad: Určení primárního klíče u entitní množiny Dodavatel. Jako kandidátní klíče jsou zvoleny atributy id dodavatele, IČO a DIČ. Atribut DIČ splňuje jedinečnost, ale v případě,

že dodavatel není plátce DPH je hodnota atributu NULL (nepřítomnost hodnoty). Atribut IČO splňuje podmínku jedinečnosti, ale při nákupu zboží od fyzické osoby, která nevlastní živnostenský list, obsahuje tento atribut také hodnotu NULL (nepřítomnost hodnoty). Jediný atribut id dodavatele splňuje všechny požadavky na primární klíč. Atribut id dodavatel se stává primárním klíčem tabulky Dodavatel.

Tento proces se aplikuje na všechny tabulky. Výsledek tohoto procesu je znázorněn do E-R diagramu (Obrázek 13). Primární klíče jsou zvýrazněny podtržením atributu.



Obrázek 13 - Praktická část, primární klíče

4.7 Normalizace

Tento krok ověří návrh pomocí normálních forem. Jelikož jde o návrh datového skladu, je doporučené postupovat do zajištění Boyce-Coddovy normální formy.

Demonstrace normalizace na tabulce Dopravce (id dopravce, název, telefon, email, cena za km):

První normální forma:

Atributy v tabulce Dopravce jsou atomické. Při aplikaci 1NF je nutné se zamyslet nad požadovaným stupněm nedělitelnosti hodnot atributů. Samozřejmě je možné dělit atribut telefon na atributy předvolba a telefonní číslo. Stejný případ je u atributu email, který jde rozdělit na atributy název a emailová doména. Je tedy potřeba rozlišovat mezi možným a požadovaným rozkladem. Po splnění tohoto předpokladu tabulka Dopravce splňuje 1NF.

Druhá normální forma:

Tabulka Dopravce nemá složený primární klíč, tím je zajištěno, že všechny neklíčové atributy jsou plně závislé na primárním klíči, a proto je tabulka v 2NF.

Třetí normální forma:

Atributy, které nejsou primárním klíčem, nejsou tranzitivně závislé na primárním klíči. Toto tvrzení by nebylo pravdivé, kdyby atribut cena za km byl závislý na atributu název dopravce a zároveň atribut název dopravce závislý na primárním klíči id dopravce. Tato vlastnost neplatí, proto je tabulka v 3NF.

Boyce-Coddova normální forma:

Tabulka vyhovuje Boyce-Coddově normální formě. Atributy, které jsou součástí primárního klíče, jsou vzájemně nezávislé.

4.8 Zajištění integrity

V tomto kroku je nutné projít čtyři typy integrity a ověřit, nebo případně demonstrovat, jejich splnění na návrhu.

Integrita na úrovni tabulky (integrita entity)

Tento typ integrity zajišťuje aplikace normálních forem na model. Tímto nevznikají duplicity a každý záznam je jednoznačně identifikovatelný pomocí primárního klíče.

Integrita na úrovni pole (doménová integrita)

Všechny atributy jsou zařazeny do domény. Doména definuje, jakých hodnot může atribut nabývat.

Objednávka:

číslo objednávky (INT, PK, NOT NULL), datum uzavření objednávky (DATE), termín dodání (DATE), množství (INT), kontaktní osoba (CHAR(30)), doba splatnosti (INT), měna (CHAR(20))

Slevová skupina:

id slevová skupina (INT, PK, NOT NULL), označení slevové skupiny (CHAR(20)), násobitel (FLOAT), podmínka zařazení (CHAR(20))

Zboží:

inventární číslo (INT, PK, NOT NULL), název zboží (CHAR(20)), cena (FLOAT), jednotky (INT), stav (CHAR(20)), popis (CHAR(20))

Dodavatel:

id dodavatel (INT, PK, NOT NULL), název dodavatele (CHAR(20)), telefon (INT), ulice (CHAR(20)), město (CHAR(20)), popisné číslo (INT), email (CHAR(40)), bankovní spojení ((CHAR(30))), IČO (INT), DIČ (CHAR(12)), právní osobnost (CHAR(20), povolené hodnoty – fyzická osoba / právnická osoba)

Odběratel:

id odběratel (INT, PK, NOT NULL), název odběratele (CHAR(20)), telefon (INT), ulice (CHAR(20)), město (CHAR(20)), popisné číslo (INT), email (CHAR(40)), právní osobnost (CHAR(20), povolené hodnoty – fyzická osoba / právnická osoba), odvětví podnikání (CHAR(20))

Kategorie zboží:

id kategorie (INT, PK, NOT NULL), název (CHAR(20)), popis (CHAR(20))

Sklad:

id sklad (INT, PK, NOT NULL), kapacita skladu (INT), ulice (CHAR(20)), město (CHAR(20)), popisné číslo (INT)

Dopravce:

id dopravce (INT, PK, NOT NULL), název dopravce (CHAR(20)), cena za km (FLOAT), telefon (INT), email (CHAR(20))

Bezpečnostní skupina ADR:

id bezpečnostní skupina ADR (INT, PK, NOT NULL), název (CHAR(20)), popis (CHAR(20))

Popis použitých zkratk:

INT – celé číslo, CHAR(20) – pole znaků o velikosti 20 (možnost použít datový typ VARCHAR), FLOAT – desetinné číslo, DATE - datum

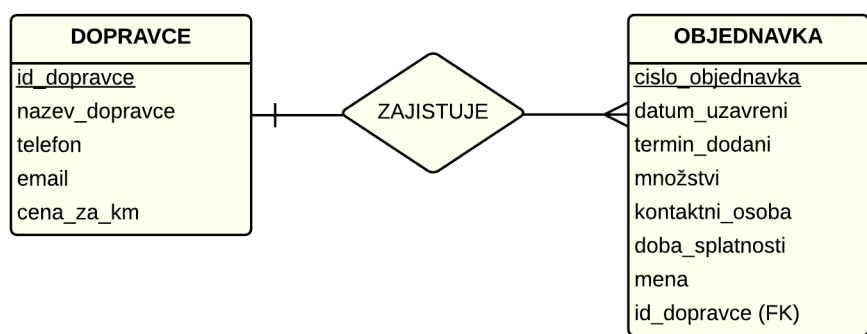
PK – primární klíč

NOT NULL – hodnota atributu nesmí nabývat NULL (nepřítomnost hodnoty). Atributy, které nejsou takto označeny, mohou nabývat hodnoty NULL

povolené hodnoty – fyzická osoba / právnická osoba – atribut může nabývat jen těchto dvou přípustných hodnot

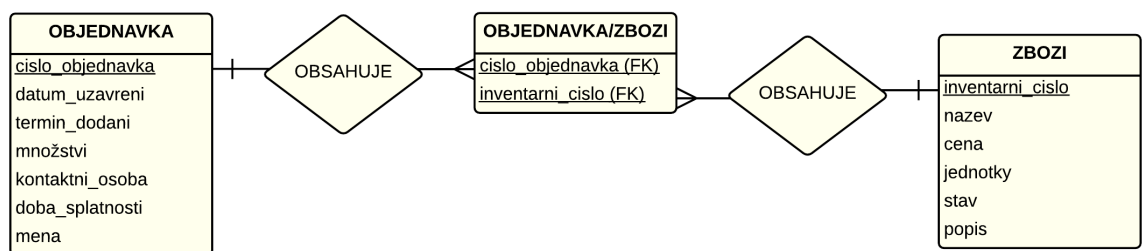
Integrita na úrovni vztahů (referenční integrita)

Referenční integrita je zajištěna primárními a cizími klíči. U relace mezi dvěma tabulkami s kardinalitou 1:N obě tabulky obsahují primární klíč a tabulka, pro kterou platí N výskytů, obsahuje cizí klíč (označený v E-R diagramu jako FK, viz Obrázek 14) na tabulku druhou.



Obrázek 14 - Praktická část, referenční integrita 1:N

V případě existence kardinality M:N vzniká vazební tabulka. Vazební tabulka obsahuje primární klíč ve formě složeného primárního klíče (Obrázek 15) nebo se vytvoří nový identifikátor. Vazební tabulka obsahuje dva cizí klíče, které propojují původní tabulky a kardinalitu M:N rozdělí na 1:N a N:1 (Obrázek 15). Tímto se zajistí konzistentnost při práci s databází.



Obrázek 15 - Praktická část, referenční integrita M:N

Business pravidla

Na navrhovaném příkladu je stanoveno business pravidlo pro tabulku Objednávka. U každého záznamu nesmí být hodnota atributu termín dodání starší než hodnota atributu datum uzavření.

4.9 Dokumentace

Praktická část této práce přímo obsahuje jednotlivé fáze, které jsou předmětem dokumentace. Reálná dokumentace by obsahovala celou kapitolu 4, včetně dokumentace,

kteřá je popsána v této kapitole. V následující části je zachycena definice významu jednotlivých tabulek a jejich atributů.

Dodavatel

Tabulka dodavatel slouží k evidenci požadovaných atributů o dodavatelích. Jako primární klíč je zvolen atribut id dodavatele. Atribut název eviduje název firmy v případě právnické osoby a v případě fyzické osoby eviduje příjmení. Atribut právní podnikání má definovány dvě hodnoty, kterých může nabývat, PO (právnická osoba) nebo FO (fyzická osoba). Ostatní atributy jasně svým názvem určují obsah dat.

Objednávka

Tabulka obsahuje primární klíč atribut číslo objednávky, který je pro každou objednávku unikátní. Atribut datum uzavření vyjadřuje datum, kdy byla objednávka přijata do systému. Atribut měna určuje, v jaké měně bude transakce provedena. Atribut kontaktní osoba je připraven pro následné rozšíření návrhu (požadovaný návrh se zabývá jen objednávkovým systémem, ne personalistikou). Ostatní atributy jasně svým názvem určují obsah dat.

Slevová skupina

Jako primární klíč je určen atribut id slevová skupina. Atribut označení vyjadřuje procentuální slevu na objednávku (např. 5%). Atribut násobitel vyjadřuje desetinné číslo, na základě požadované slevy (požadovaná sleva: 5% - hodnota atributu násobitel: 1-0,05). Ostatní atributy jasně svým názvem určují obsah dat.

Dopravce

Pod tabulkou dopravce se rozumí způsob dopravy. Primárním klíčem je id dopravce. V tabulce je možné uchovávat záznamy o externích dopravcích, kteří zajišťují dopravu, ale i způsob přepravy ve formě osobního převzetí, při kterém jsou atributy telefon a email NULL. Ostatní atributy jasně svým názvem určují obsah dat.

Odběratel

Tabulka odběratel slouží k evidenci požadovaných atributů o odběratelích. Jako primární klíč je zvolen atribut id odběratele. Atribut název eviduje název firmy v případě právnické osoby a v případě fyzické osoby eviduje příjmení. Atribut právní podnikání má definovány dvě hodnoty, které může nabývat, PO (právnická osoba) nebo FO (fyzická osoba). Ostatní atributy jasně svým názvem určují obsah dat.

Bezpečnostní skupina ADR

Tabulka bezpečnostní skupina ADR vyjadřuje český ekvivalent tohoto termínu (Evropská dohoda o mezinárodní silniční přepravě nebezpečných věcí). Jako primární klíč je zvolen atribut id bezpečnostní skupina. Atribut název vyjadřuje zkratku dané bezpečnostní skupiny určenou podle zákona. Ostatní atributy jasně svým názvem určují obsah dat.

Zboží

U tabulky zboží je za primární klíč určen atribut inventární číslo, které je jedinečné pro každý záznam. Atribut jednotky určuje jednotky, ve kterých je zboží evidováno (jelikož jde o návrh pro zadavatele obchodujícího s plyny, tento atribut může nabývat hodnot ks, m³, litry atd.). Atribut stav vyjadřuje skupenství daného zboží (plynné, kapalné). Ostatní atributy jasně svým názvem určují obsah dat.

Kategorie

Jednoduchá tabulka, která obsahuje jen primární klíč a název skupiny a popis. Jedná se o kategorie zboží (např. technické plyny, speciální plyny, chladiva, plyny využívané ve zdravotnictví, sloučeniny do klimatizací atd.)

Sklad

Tabulka evidující sklady, které zadavatel vlastní. Atribut kapacita skladu vyjadřuje počet metrů čtverečních pro skladování standardizovaných palet s tlakovými lahvemi s plynem. Ostatní atributy jasně svým názvem určují obsah dat.

4.10 Budoucí vývoj

Návrh databáze pro evidenci objednávek je dokončen. Za budoucí vývoj je možné považovat rozšíření databáze do oblasti personalistiky. K tomuto je zde připraven atribut kontaktní osoba v tabulce Objednávka. Atribut kontaktní osoba se může stát entitní množinou a být doplněn novými atributy a následně navázán na další tabulky. Návrh nemusí být v případě nového požadavku (např. přidání atributu) celý předělán, ale stačí jednoduchá modifikace.

5 Výsledky a diskuze

Práce se zaměřuje na problematiku relačních databází a objasňuje základní terminologii této oblasti. V práci je autorem navržen postup návrhu databázové základny, který je odvozen z literárních zdrojů doplněných o poznatky autora.

V praktické části práce autor reaguje na výsledky analýzy, které jsou součástí definovaného postupu návrhu datové základny. V analýze je použita SWOT analýza, jejíž výsledkem je určení silných a slabých stran dosavadního řešení, na které autor bere zřetel při návrhu datové základny. Dále analýza definuje příležitosti, ze kterých autor těží při návrhu a hrozby, kterým se snaží vyvarovat. Vlastní návrh klade hlavní důraz na předcházení datových redundancí při následné implementaci. Tato potřeba vychází z analýzy, kde jsou zjištěny nedostatky jako zbytečné atributy v tabulkách. Návrh bere v potaz část analýzy, ve které jsou vyhodnoceny rozhovory se zadavatelem, managementem a uživateli. Na základě těchto rozhovorů se jsou určeny nadbytečné atributy, které nové řešení ignoruje. Vlastní řešení se dále zaměřuje na problém nejednoznačného pojmenovávání tabulek a atributů v dosavadním řešení a správného utčení všech primárních klíčů. Tento problém je vyřešen na základě autorem definovaného postupu, který obsahuje doporučení pro eliminování těchto problémů. Následně autor zdůrazňuje důležitost vytvoření dokumentace k návrhu, která slouží zaprvé jako vodítko pro vlastní implementaci a zadruhé jako shrnutí všech kroků návrhu pro další možnou modifikaci. V závěru procesu návrhu je zhodnocen možný budoucí vývoj.

Jedná-li se o splnění cílů definovaných u vlastního navrhovaného řešení, je možné konstatovat, že byly splněny. Po skončení procesu návrhu je výsledný model připravený k implementaci. Vlastní návrh obsahuje oblasti, které byly definovány jako cíle návrhu, tedy práce s objednávkami, evidování informací o dodavatelích a odběratelích a informace o zásobách zboží.

Autor podotýká, že vytvořit vlastní řešení je až druhořadou prioritou. Jak již vychází z názvu práce je hlavní prioritou definovat obecný postup, kterým se může řídit jakýkoliv návrh datové základny bez ohledu na problematiku, kterou se zabývá. Je třeba brát v zřetel, že vlastní proces návrhu v praxi vykonává tým specialistů, a proto zde uvedený vlastní návrh nemusí být optimálním řešením, ale složí k demonstraci obecného postupu navrženého autorem.

6 Seznam použitých zdrojů

6.1 Monografické zdroje

[1] HERNANDEZ, M.J.: Návrh databází. Grada Praha 2006. ISBN 9788024709000, 8024709007

[2] LACKO, L.: ORACLE. Správa, programování a použití databázového systému. Computer Press Brno 2010. EAN 97880251149002

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

[4] VOSTROVSKÝ, V.: Vytváření databází v Oracle. ČZU v Praze. ISBN: 978-80-213-1191-6

6.2 Internetové zdroje

[5] Databaze.chytrak. [online]. 2010 [2014-08-29] Dostupné na: www.databaze.chytrak.cz

[6] en.wikipedia.org . [online]. 2014 [2015-03-04] Dostupné na: http://en.wikipedia.org/wiki/Boyce%E2%80%93Codd_normal_form

[7] TryHana. [online]. 2012 [2015-01-20] Dostupné na: <http://www.tryhana.cz/SAP-Hana.aspx>

7 Seznam použitých obrázků

Obrázek 1 - Hierarchie abstrakcí a jejich realizace (zdroj: POKORNÝ, J.: Databázové systémy. ČVUT Praha 2013. s. 18)	15
Obrázek 2 – Chenova notace, relace 1:N.....	17
Obrázek 3 - Výlučné vztahy v binárním modelu (zdroj: POKORNÝ, J.: Databázové systémy. ČVUT Praha 2013. s. 26).....	18
Obrázek 4 - Agregace UML (zdroj: POKORNÝ, J.: Databázové systémy. ČVUT Praha 2013. s. 28).....	18
Obrázek 5 - Grafické znázornění SWOT analýzy	22
Obrázek 6 - Vztah 1:1 (zdroj: LACKO, L.: ORACLE. Správa, programování a použití databázového systému. Computer Press Brno 2010. s. 17)	24
Obrázek 7 - Vztah 1:N (zdroj: LACKO, L.: ORACLE. Správa, programování a použití databázového systému. Computer Press Brno 2010. s. 17)	24
Obrázek 8 - Vztah N:M (zdroj: LACKO, L.: ORACLE. Správa, programování a použití databázového systému. Computer Press Brno 2010. s. 18)	25
Obrázek 9 - Vztah N:M, rozložení pomocí vazební tabulky (zdroj: LACKO, L.: ORACLE. Správa, programování a použití databázového systému. Computer Press Brno 2010. s. 18)	25
Obrázek 10 - Unární relace (zdroj: LACKO, L.: ORACLE. Správa, programování a použití databázového systému. Computer Press Brno 2010. s. 18)	26
Obrázek 11 - Grafická reprezentace obecného normalizačního procesu, [1, str 61]	27
Obrázek 12 - Praktická část, určení relací	38
Obrázek 13 - Praktická část, primární klíče.....	41
Obrázek 14 - Praktická část, referenční integrita 1:N.....	45
Obrázek 15 - Praktická část, referenční integrita M:N	45

8 Seznam použitých tabulek

Tabulka 1 - Nultá normální forma	28
Tabulka 2 - První normální forma před úpravou	28
Tabulka 3 - První normální forma po úpravě	29
Tabulka 4 - Druhá normální forma před úpravou	29
Tabulka 5 - Druhá normální forma po úpravě 1	30
Tabulka 6 - Druhá normální forma po úpravě 2	30
Tabulka 7 - Třetí normální forma před úpravou	30
Tabulka 8 - Třetí normální forma po úpravě 1	31
Tabulka 9 - Třetí normální forma po úpravě 2	31