

ÚVOD DO KONCEPTUÁLNÍ NORMALIZACE

Martin Molhanec

České vysoké učení technické v Praze

Fakulta elektrotechnická, K13113

Technická 2, 166 27 PRAHA 6, Dejvice, Česká republika

tel.: (++420) 2 2435 2118

mailto: molhanec@fel.cvut.cz

http://martin.feld.cvut.cz/~molhanec

ABSTRAKT.

Tento příspěvek se zabývá některými úvahami o datové normalizaci v konceptuální oblasti, a jak z této normalizace vyplývá normalizace relační i objektová. Jedná se pouhý úvod do zajímavé a svým způsobem zásadní problematiky, která je však opomíjená. Příspěvek nepodává ucelený a formální aparát, ani přesné matematické důkazy jednotlivých tvrzení, jeho cílem jest stát se spíše podkladem pro další práci v této problematice a počátečním bodem diskuze na poli odborné veřejnosti.

KLÍČOVÁ SLOVA: normalizace datová, konceptuální, relační a objektová, normální formy, konceptuální modelování

1 ÚVOD

Již delší dobu se ve své vědecké a odborné práci zabývám problematikou konceptuální normalizace. Této opravdu zásadní problematiky jsem se dotkl v některých svých minulých příspěvcích na konferenci Objekty [6, 5, 9] či na tomto semináři [8 a 10]. Tento příspěvek je zamýšlen jako velice stručný nástin některých tvrzení a myšlenek, ke kterým jsem v posledních letech dospěl. Nebude proto obsahovat ucelený formální aparát, ale spíše neformální tvrzení a definice, ani nebude obsahovat přesvědčivé a definitivní důkazy jednotlivých tvrzení. Doufám, že může být počátečním bodem pro zajímavou a podnětnou diskuzi odborníků v této oblasti a pochopitelně východiskem mé další vědecké práce.

2 MOTIVACE

Důvod toho, že se zabývám konceptuální normalizací, vyplývá z jedné zásadní skutečnosti. Normalizace v relační oblasti (1. až 3. NF, a další NF) je běžně zavedená skutečnost, která se učí v každém vysokoškolském databázovém kursu a nikdo nepochybuje o její užitečnosti a oprávněnosti. Bohužel v oblasti objektově orientovaného paradigmatu jsem se setkal mnohdy s tvrzeními, že normalizace je pro oblast objektů nesmysl, protože v objektových databázích nejsou přeci žádné primární ani cizí klíče. Jak nesmyslné tvrzení! Pochopitelně, že v objektové databázi klíče nepotřebujeme, ale samotná normalizace jest proces, jak zbavit náš datový model nadbytečnosti a definice normálních forem pro relační databáze s pomocí klíčů je pouze vhodné vyjádření tohoto faktu v relační oblasti. O tom, že objektová

normalizace je skutečností, svědčí jednak celá řada příspěvků v zahraničí, ale i v tuzemsku, např. [1] a [5]. Je tu však několik problémů:

- Na rozdíl od relační oblasti není žádná definitivní a všeobecně přijatá definice objektových normálních forem.
- Většina autorů má problém, jak v definicích nahradit koncept klíče z relační oblasti jiným vhodným konceptem.
- Není ani jasné zdali je v objektové oblasti stejný počet normálních forem jako v oblasti relační.

Ve svých úvahách vycházím proto z následujícího předpokladu:

Jak relační, tak objektový model, jsou specializací daleko obecnějšího modelu konceptuálního, proto jak relační, tak objektová normalizace, jsou taktéž specializací normalizace konceptuální. (1)

Považuji za potřebné zmínit v tomto místě jedno zavádějící tvrzení, se kterým se často setkávám, při hovorech o objektové normalizaci a rozdílech mezi objektovým a relačním přístupem.

** Objektový přístup nemá a nepotřebuje žádné primární a cizí klíče, proto je lepší než přístup relační a jeho normalizace je tedy nesmyslná. (2)*

Toto tvrzení spolu s představou, že normalizace je něco co souvisí s klíči, implikuje právě totiž představu, že v objektové oblasti je normalizace něco nesmyslného. Jak jsem však již předeslal, normalizace není o klíčích, ty jsou jen v relační oblasti určitým vhodným nástrojem, jak normální formy v této oblasti explicitně definovat, ale o nadbytečnosti, která vyvěrá z neexistence nadbytečnosti v reálném světě.

Navíc, pokud definice normálních forem přeformulujeme na bázi funkčních závislostí, případně pro objektovou oblast s využitím interních identifikátorů objektů, dostáváme se nakonec více či méně k témuž paradigmatu.

Výše uvedené tvrzení navíc vede ke zlozvyku některých objektových programátorů neuvažovat při návrhu objektové databáze o způsobu jednoznačné identifikace objektů na základě jejich skutečných atributů, která je však nutná čistě z praktických důvodů. I v objektové oblasti je totiž nutné vědět, jak daný objekt jednoznačně vyhledat na základě znalostí hodnot jeho atributů, případně jak se vyhnout nežádoucímu vzniku duplicitního objektu při jeho vytvoření. A tyto úvahy je pochopitelně žádoucí provést ihned na počátku vývojového procesu celé aplikace, nežli dodatečně až když nás k tomu okolnosti jejího praktického užívání donutí.

V dalším textu si ukážeme, jak lze konceptuální normalizaci definovat a odvodit z ní normalizace pro obě dvě oblasti našeho zájmu – oblast relační i oblast objektovou. Budeme však muset překonat ještě jednu zásadní potíž – podobně jako v oblasti objektové nemáme k dispozici žádné klíče, na kterých bychom snadno své definice normálních forem uskutečnili.

3 KONCEPTUÁLNÍ MODEL

První věc, kterou je nutné si uvědomit, že konceptuální model nemá nic společného s výpočetní technikou, programováním a databázemi, jedná se totiž, jak ostatně již z jeho názvu vyplývá, o model pojmový čili model pojmů/konceptů vyskytujících se v reálném světě okolo nás. Konceptuální model je výsledkem ontologického nazírání na svět a vyplývá z ontologického modelu světa kolem nás. Ve svých dalších úvahách budu, aniž bych musel zabíhat do oblasti ontologie, však pracovat s běžným konceptuálním modelem, který je velice blízký modelu objektovému. Základní konstrukty tohoto modelu jsou:

- Objekt
- Třída
- Dědičnost
- Atribut
- Skládání
- Vztah

Přestože se na první pohled tento model zdá totožný s modelem objektovým není tomu tak. Nesmíme o tomto modelu anticipovat žádné vlastnosti, které souvisejí s implementací objektového modelu v počítači nebo vyplývající z toho kterého objektového programovacího jazyka.

Neexistuje zde žádný jedinečný interní identifikátor objektu, na rozdíl například od objektových systémů. (3)

Toto tvrzení jest triviální. Opravdu nelze tvrdit, že objekty v reálném světě mají od přírody nějaký interní identifikátor. Jedinečná identifikace objektů, jako například výrobní číslo výrobku je vlastnost uměle vytvořená člověkem právě za účelem jednoznačné identifikace. Ale většina objektů reálného světa nic takového nemá. Ano, u člověka by v úvahu mohla přicházet například jednoznačná genetická informace, ale v tomto případě se jedná spíše o jednoznačnou a identifikující strukturu, než o jednoduchý jedinečný identifikátor, který máme na myslí.

Neexistují totožné třídy z hlediska své intenze. (4)

I toto tvrzení si vyžaduje krátké vysvětlení. Na rozdíl od objektových programovacích jazyků vzniká třída (*class*) v reálném světě odlišným způsobem. V běžném objektovém jazyce můžeme vytvořit dvě třídy, které mají stejnou intenzi, tj. jsou definovány stejnou množinou

atributů a metod a přesto je systém bude chápat jako dvě odlišné třídy, ač z hlediska jejich chování a vlastností budou vlastně totožné. V reálném světě žádné třídy vlastně neexistují (pokud odhlédneme od Platónovy koncepce světa idejí). Třída v reálném světě je pojem, který vyplývá z abstraktní schopnosti člověka chápat a uvědomovat si, že určité množiny objektů jsou si sobě podobné. Proto je třída v konceptuálním modelu definována jedine svojí intenzí, extenzi není totiž možné plně obsáhnout, a jediné co dokážeme je, že dané intenzi přidělíme různé názvy (nálepky/označení), ale i tak stále jest pouze jediná třída příslušející k dané intenzi, pro kterou však máme více jmen, tj. synonym.

3.1 Základní tvrzení o konceptuální normalizaci

Předpoklad, ze kterého ve všech dalších úvahách vycházím je zcela triviální.

Reálný svět není nadbytečný. (5)

Uvědomme si následující skutečnost. Všechny objekty v našem reálném světě existují opravdu pouze jedinkrát. Existuje pouze jeden *Martin Molhanec*, existuje pouze jeden výskyt konference *Objekty 2009*, atp. Nebudu se zde pouštět do pochybných filozofických úvah o alternativních světech. V reálném světě prostě neexistují dvě totožné instance téhož objektu v témže čase a prostoru. Naše informační systémy ve své datové formě obsahují model reálného světa, je tedy zřejmé, že podobně jako je svět bez redundancí, tak i model tohoto světa v daném informačním systému by měl být bez nich taktéž. Pochopitelně, nemluvím zde o datových skladech, které z určitých důvodů vytvářejí redundanci za účelem dosažení svých specifických vlastností. Zjednodušeně lze tedy tvrdit, že podobně jako je reálný svět bez nadbytečnosti, tak ani naše databáze neměla by nadbytečnosti obsahovat. Pochopitelně, že praxe pak tuto zásadu často vědomě opouští především s cílem zvýšení výkonu informačního systému při databázových operacích.

3.2 Konceptuální objekt a jeho vlastnost

Prozatím jsme se v našem textu nezabývali vlastnostmi konceptuálních objektů. Vyjděme z jednoduché přirozené definice:

Vlastnost je určitá charakteristika objektu, která má svojí hodnotu. (6)

Příkladem takové vlastnosti může být třeba: barva nebo věk. Obecně může vlastnost být i množina hodnot. A úplně nejobecněji může být vlastností i vztah jednoho objektu ke druhému. Vlastnost je určitá abstrakce charakterizující daný objekt, součást jeho intenze, kterou člověk rozpoznává/vnímá. Pro vztah mezi objektem a jeho vlastnostmi platí následující triviální tvrzení:

Množina vlastností v rámci objektu je jedinečná. (7)

A skutečně. Auto má jenom jednu svou vlastnost barva (pochopitelně barva karoserie a barva například podvozku jsou dvě rozdílné vlastnosti). Osoba má pouze jedinou vlastnost věk, atp. Důkaz výše uvedeného tvrzení je možné provést s ohledem na skutečnost, že vlastnost je v rámci daného objektu opět konceptem a tedy abstraktním objektem a tedy i pro ni pak platí tvrzení o jedinečnosti objektů v reálném světě. Auto nemá současně v dané souřadnici časoprostoru dvě barvy nebo osoba dva věky.

S vlastností objektů pak souvisejí následující tvrzení.

Vlastnost objektu není dělitelná, je atomická. Pokud potřebujeme pracovat s její částí, stává se sama objektem, často abstraktním, který má své vlastní vlastnosti. (8)

Pokud potřebuji pracovat se skupinou vlastností objektu, jako s jedním pojmem, je tento pojem sám objektem, často abstraktním. (9)

Potenciální atomičnost vlastnosti je doménově závislá. (10)

Vysvětleme si, jak výše uvedená tvrzení chápat na jednoduchém příkladu se jménem osoby. Pokud v zájmové doméně vždy pracujeme se jménem jako vlastností, která v sobě zahrnuje jak jméno křestní, tak příjmení, je vše v pořádku a můžeme tuto vlastnost chápat jako atomickou a jedinečnou v rámci pojmu osoba. Pokud v rámci dané domény cítíme potřebu pracovat se jménem křestním a příjmením zvlášť, pak se jméno samo stává abstraktním objektem, a má dvě vlastnosti. A naopak, pokud má objekt osoba vlastnosti: jméno křestní a příjmení, a v rámci dané domény máme potřebu pracovat s dvojicí těchto vlastností současně jako s jediným pojmem, stává se tato dvojice sama o sobě pojmem/objektem se svým názvem/nálepku jméno.

Navíc nesmíme zapomenout, že v konceptuálním světě je možné na vlastnost zásadně vždy pohlížet duálním způsobem.

Vlastnost objektu lze chápat současně jako jiný objekt ve vztahu k tomuto objektu. (11)

Z této duality a z různých úhlů pohledu (domén) vyplývá následně i důvod, proč, přestože je jediná existující časoprostorová realita, jí můžeme modelovat mnoha různými a vždy správnými způsoby.

3.3 Vlastnost, objekt a třída

V našem textu jsme doposud nehovořili o definici konceptu třídy. Zavedeme si ji tedy následujícím způsobem. (Připomínám, mluvíme zde o konceptuální třídě, nikoliv o třídách implementačních/programátorských.)

Třída je pojmenovaná určitá množina vlastností náležících sobě podobným objektům. (12)

Je nutné si uvědomit zásadní věc. V reálném světě je třída abstraktní koncept, který nám pojmenovává množinu reálných i abstraktních objektů, které chápeme jako sobě podobné. Třída auto je tedy pojmenování množiny objektů, které mají takové vlastnosti, že je chápeme, jako něco navzájem podobného, co si zaslouží vlastní hromadné pojmenování a tímto pojmenováním je název třídy. Nechci zabíhat do hloubky tohoto konceptu, který velice úzce souvisí jak s filosofií, tak s lingvistikou. Bohužel i díky tomu se velice špatně formuluje v běžné řeči neformálním a snadno srozumitelným způsobem.

Definujme nyní další dva pomocné pojmy:

Intenze třídy je množina jejích vlastností. (13)

Extenze třídy je množina všech objektů dané třídy. (14)

Na rozdíl od světa programátorů, kde si snadno definujeme intenzi třídy a pak vytváříme její extenzi. V reálném světě, kde jsou pouze reálné objekty, musíme na základě studia množiny sobě podobných objektů definovat nejprve její extenzi, která nás zajímá a na základě studia vlastností objektů náležících do této extenze definovat pro naši doménu její potenciální intenzi.

3.4 Objekt, vlastnost, třída a vztah

Doposud jsme v našem textu příliš nehovořili o vztazích. Vycházejíce z ontologického přístupu [3] rozeznáváme v našem konceptuálním modelu následující 4 druhy vztahů:

- Vztah mezi objektem a jeho vlastností.
- Vztah dědičnosti mezi třídami.
- Vztah skládání objektů.
- Obecný vztah mezi objekty.

Všechny tyto vztahy jsou obecně v modelovacích technikách používané, snad až na první vztah, který se jaksí předpokládá implicitně. Je nutné si však uvědomit následující skutečnost: Přestože obvykle všechny výše jmenované skutečnosti nazýváme slovíčkem/nálepkou vztah nemají mezi sebou vlastně nic společného, není vlastně žádný důvod pro jejich společné pojmenování. V podstatě se jedná o rozdílné skutečnosti a jejich hromadné pojmenování slovem/nálepkou vztah může být ve svém důsledku nakonec zavádějící.

Na druhou stranu je možné, jak bylo již výše naznačeno, chápat vztah mezi objektem a jeho vlastností nebo vztah mezi více objekty za dva podtypy vztahu obecnějšího, kdy vztah mezi

objektem a jeho atributem je případem, ve kterém tento vztah se uskutečňuje jako singleton a je svou podstatou vztahem atributárním, zatímco běžný vztah se uskutečňuje mezi nejméně dvěma objekty (vztah n-ární). Toto pojení je v souladu s ontologií GOL [3].

Vztah dědičnosti se chápe obvykle jako vztah mezi třídami. Ontologicky chápaný v konceptuálním modelu, jako vztah vyvěrající z podobnosti reálných objektů, jak bylo zmíněno již výše.

Podobně vztah skládání vyplývá z reálné možnosti fyzických objektů tvořit struktury. Samotná složenina může být pak pochopena jako vlastnost objektu. Podle úhlu pohledu (domény) zejména v abstraktním světě, ale i v reálném, můžeme však určitý vztah mezi objekty chápat buď jako vztah skládání nebo obecný vztah. Problematikou přesného určení, které vyplývá z ontologie, o jaký typ vztahu se jedná, se zabývá velké množství teoretických prací, např. [3]. Často se při určení, toho o jaký typ vztahu se vlastně jedná, používají různé pomocné otázky, například „Může existovat složenina beze všech svých prvků?“, atp.

4 KONCEPTUÁLNÍ NORMÁLNÍ FORMY (CNF)

Jako konceptuální normální formy (CNF), tj. pravidla o zákazu nadbytečnosti chápu svá tvrzení 5, 7, 8 a 9 výše uvedená v textu, která zde opět pro připomenutí zopakují.

0. CNF: Reálný svět není nadbytečný. (5)

1. CNF: Množina vlastností v rámci objektu je jedinečná. (7)

2. CNF: Vlastnost objektu není dělitelná, je atomická. Pokud potřebujeme pracovat s její částí, stává se sama objektem, často abstraktním, který má své vlastní vlastnosti. (8)

3. CNF: Pokud potřebuji pracovat se skupinou vlastností objektu jako s jedním pojmem. Je tento pojem sám objektem, často abstraktním. (9)

Zdůvodnění proč považují tato tvrzení za základní tvrzení o normalizaci, tj. tvrzení o zákazu nadbytečnosti byla uvedena v podstatě v předešlých kapitolách hned s uvedením těchto tvrzení. Domnívám se, že definice normálních forem, jak v relační, tak v objektové formě, lze z těchto konceptuálních normálních forem odvodit. Pokusím se proto v dalším textu tato odvození aspoň neformálně naznačit.

4.1 Relační normální formy (RNF)

Jsem přesvědčen, že 1. RNF vyplývá z výše definované 0. až 2. CNF. Mé přesvědčení vyplývá ze skutečnosti, že se 1. RNF zabývá atomicitou datových atributů, zákazem multiatributů a potřebností existence primárního klíče. Atomicitou datových atributů se zabývá zde definovaná 2. CNF, zákazem multiatributů 1. CNF a konečně nutnost existence primárního klíče vyplývá ze základní 0. CNF.

Dle mého přesvědčení 2. RNF je ve své podstatě pouze specializací obecnějšího principu 3. RNF (s nutností určitého přeformulování této RNF), ale rozsáhlejší diskuzi tohoto tvrzení s ohledem na rozsah tohoto příspěvku zde uvádět nebudu. Potom jak 2., tak i 3. RNF plynou ze zde navržené definice 3. CNF.

4.2 Objektové normální formy (ONF)

Bohužel neexistuje obecně přijatý standard pro definice objektových normálních forem. Existuje celá řada příspěvků na toto téma, přehled některých přístupů je uveden v [5]. V této práci jsou také definovány objektové normální formy vycházející z koncepce Scotta Amblera [1] a dále rozpracované autory Merunkou a Molhancem. Přestože autoři definují normální formy pro objektovou oblast, jejich definice vycházejí z definic jim odpovídajících relačních norem, zbavených však jakýchkoliv odkazů na konstrukt relačního klíče, který nám v objektové doméně chybí. Nicméně protože jednotlivé objektové formy ve zmíněné práci definované odpovídají jednotlivým tradičním relačním normálním formám, je zřejmé, že možnost jejich odvození z konceptuálních norem v tomto příspěvku definovaných, je stejná jako pro obvyklé relační normy, jak je uvedeno v předcházející části příspěvku (4.1).

5 ZÁVĚR

Jsem toho názoru, že skutečnost, že jak relační, tak objektová normalizace, vyplývá ze stejných zdrojů, tedy z normalizace konceptuální, není nic překvapujícího. Je jen s podivem, že jen málo autorů se touto problematikou zabývá a že není více známa odborné veřejnosti, zejména objektově orientované. Autorova „Krátká úvaha o normalizaci“ zde předkládaná, je jen stručný vchod do celé problematiky. Je totiž nutné všechny definice v konceptuální oblasti dostatečně pevně ontologicky ukotvit a vypracovat formální definice normalizací ve všech třech studovaných oblastech včetně jejich transformací z konceptuální oblasti do oblasti relační i objektové.

Pochopení normalizace jak v relační, tak v objektové oblasti je nesmírně důležité pro návrh bezpečných a robustních informačních a databázových systémů. I porušování pravidel normalizace, za účelem zvýšení jejich výkonu či vytvoření datových skladů, musí být vedeno se zámyslem úmyslného a vědomého jejich porušení metodicky kompenzovaného užitím triggerů nebo jiných programovacích technik.

Autor příspěvku doufá, že tímto svým příspěvkem vyvolá diskuzi a zájem čtenářů o tuto zajímavou problematiku.

V Praze 3. dubna 2010
Martin Molhanec

Poděkování

Tato práce je podporována výzkumným záměrem MSM6840770017 Ministerstva školství, mládeže a tělovýchovy ČR.

LITERATURA

1. Ambler Scott: *Building Object Applications That Work, Your Step-By-Step Handbook for Developing Robust Systems Using Object Technology*, Cambridge University Press/SIGS Books, 1997, ISBN 0521-64826-2
2. Edgar F. Codd, Wikipedia, online: [http://en.wikipedia.org/wiki/Edgar_F._Codd]
3. Heller, B., Herre, H. 2004. „Ontological Categories in GOL.“, *Axiomathes* 14(1):57-76 Kluwer Academic Publishers.
4. Christopher J. Date, Wikipedia, online: [http://en.wikipedia.org/wiki/Christopher_J._Date]
5. Molhanec, M., Merunka, V., Object normalization as the contribution to the area of formal methods of object-oriented database design, In: *Advances in Computer and Information Sciences and Engineering*. Berlin: Springer Science+Business Media, 2007, p. 100-104. ISBN 978-1-4020-8740-0.
6. Molhanec, M., Krátká úvaha o normalizaci, In: *Objekty 2009*. Hradec Králové: Univerzita Hradec Králové, 2009, díl 1, s. 149-160. ISBN 978-80-7435-009-2.
7. Molhanec, M., Ontologické základy konceptuální normalizace, In: *Objekty 2006*. Ostrava: Technická universita Ostrava - Vysoká škola báňská, 2006, s. 81-90.
8. Molhanec, M., Konceptuální modelování, formální základy a ontologie, In: *Tvorba softwaru 2006*. Ostrava - Poruba: VŠB - Technická univerzita Ostrava, 2006, s. 121-127. ISBN 80-248-1082-4.
9. Molhanec, Martin. „Konceptuální modelování“, In *Objekty 2005*, Ostrava: VŠB, Technická Univerzita, 2005. ISBN 80-248-0595-2.
10. Molhanec, Martin. „Zásady konceptuálního totálně objektově orientovaného modelování“ In: *Tvorba softwaru 2005*. Ostrava: VŠB, 2005, s. 153-158. ISBN 80-86840-14-X.